# Perceived Subjective Features of Software Components: Consumer Behavior in a Software Component Market

Janghyuk Lee, Se-Joon Hong, Yeong-Wha Sawng, and Ju Seong Kim

**Component-based software reuse has been generally regarded as a promising approach to improving software productivity and quality within software development. However, progress in component-based software reuse has been slower than expected. Much of the software reuse literature points to the lack of software components that can maximize users' benefits as the most important source of the slow progress. Considering that the underlying processes behind component-based software reuse are strikingly similar to commercial software marketing, this paper attempts to identify the aspects of software components that consumers value and to establish relationships between the identified aspects and consumer behavior in the software component market. More specifically, this paper focuses on the perceived subjective features of software components. This study was conducted in a web-based artificial market environment called "SofTrade."**

**Keywords: Software components, software reuse, software component market, consumer perceptions, consumer preferences, purchasing behavior.**

Manuscript received Nov. 6, 2008; revised Mar. 11, 2009; accepted Apr. 7, 2009.

Janghyuk Lee (phone: +82 2 3290 2819, email: janglee@korea.ac.kr) and Se-Joon Hong (phone: +82 2 3290 2812, email: sejoon@korea.ac.kr) are with Business School, Korea University, Seoul, Rep. of Korea.

Yeong-Wha Sawng (email: ywsong@etri.re.kr) and Ju Seong Kim (email: juskim@etri.re.kr) are with Research Strategy Laboratory, ETRI, Daejeon, Rep. of Korea

## I. Introduction

Since McIlroy [1] first envisioned the construction of complex software systems from small building blocks, component-based software reuse has been generally regarded as a promising approach to improving software productivity and quality within software development [2]. Component-based software reuse encourages the move from currently popular huge monolithic systems to modular structures (encompassing requirements, architectures, designs, and implementations) that offer the benefits of enhanced adaptability, scalability, and maintainability. These require fewer major release changes and, in turn, avoid the resulting "upgrade treadmill" of entire systems [3]. Favaro and others [4] categorize the general benefits of component-based software reuse into two categories: first, operational benefits, such as improved quality, higher productivity, and reduced maintenance costs; and second, strategic benefits, such as the opportunity to enter new markets or the flexibility to respond to competitive market forces and rapidly changing market conditions.

Despite these potential benefits, progress in component-based software reuse has been slow [5], [6]. In an attempt to understand why, researchers have studied factors that promote or impede component-based software reuse.

Although most previous studies have focused on technical problems [7], significant progress has been made in technical areas [8] so that software reuse is not radically impeded by the lack of tools and technology any more [9].

Given this progress in technical areas, more attempts to explain the slow progress of software reuse have focused on non-technical issues [6], [10]. According to Glass [6], of these non-technical issues, much of the software reuse literature points to the managerial aspect as the most important source of the slow progress of software reuse [11]. Indeed, many researchers [2], [6], [12], [13] suggest that the lack of economic incentives for component-based software reuse might be one of its main inhibitors.

To address this issue, Barnes and Bollinger [12] propose that component-based software reuse should be understood within a market context in which market participants behave in their own economic interests. From this perspective, the underlying processes behind component-based software reuse are strikingly similar to commercial software marketing. That is, producers of software components should try to develop components that satisfy consumers' needs. In fact, the fundamental problem of current component-based software reuse is that there are not enough software components that can be reused by consumers [6]. Consequently, learning what consumers consider important in evaluating and choosing software components is critical for the success of component-based software reuse. Hong and Lerch [14] explain consumers' preference and purchasing behavior with objective features of software components such as the number of lines of code and the number of methods.

In this light, the objective of this paper is to increase our knowledge of what consumers like and what they are willing to buy in the software component market, and by so doing to help producers of software components develop more marketable components that provide consumers with benefits in building applications. More specifically, this paper examines consumers' perceptions specific to component subjective features and those perceptions' impact on consumer behavior (that is, consumer preferences and purchasing behavior) in an artificial market environment called "SofTrade."

## II. Theoretical Background

In consumer behavior research, it is generally agreed that consumer decision-making processes involve careful evaluation of various product attributes [15], [16]. However, because of imperfect information and simplifications in information processing and decision rules, consumers often abstract these attributes into relatively few perceptual dimensions such as "flexibility" and "usefulness" [17]. Since these dimensions represent how consumers perceive and interpret a product's features to create a meaningful picture of the product, consumer researchers call these dimensions *perceptions* [18]. Indeed, perceptions have been popularly used for a variety of product categories, and have predicted consumer preferences and choice quite well in previous empirical studies [17].

Similarly, within information system (IS) research, a number of researchers have proposed that perceptions are critical for users to adopt and use a certain IT product [19], [20]. This perspective has resulted in a multitude of theoretical models based on various user perceptions (such as the technology acceptance model and the theory of planned behavior), and these models have been used to explain adoption and usage behavior across a wide variety of IT products, such as a smart-card systems [21] and interactive online help desk systems [22]. Most of these are intention-based models stipulating that the intention to adopt a technology is a good predictor of its actual usage. In turn, intention is explained by different perceptions.

The study reported here shares a view similar to the perspective in both consumer behavior research and IS adoption research that perceptions of consumers (users) play an important role in determining their purchasing decisions within the context of software components. That is, consumers of software components develop perceptions based on the attributes of software components, and use these perceptions to decide whether or not they will purchase them.

In addition, the current study claims the possibility of a more direct link between perceptions and purchasing behavior. It is often observed in the real world that consumers may not always choose products based primarily on their affective feelings. For example, many people like Mac OS, but they still buy Windows to be able to perform their tasks. Although in traditional consumer behavior research, consumer preferences are seen as playing a pivotal role in explaining consumer behavior, there is also a body of empirical evidence showing that people's decisions toward a behavior may not always be explained solely by their preferences [19], [23].

Since the perceptions discussed in this study are specific to software components, it is expected that they will provide valuable information to better help enhance the marketability of software components and thus improve software reuse.

## III. Artificial Software Component Market: SofTrade

Markets have been previously used productively to help determine what consumers value in most goods in our economy. From this perspective, we built an artificial software component marketplace called "SofTrade." It simulates the general features of a real market by incorporating market components and conditions like those that would be found in a real market, such as multiple participants, products, advertisements, and promotion plans. Our empirical results are, of course, bounded by the choices we made when building this market environment.

A possible concern related to this approach of using an

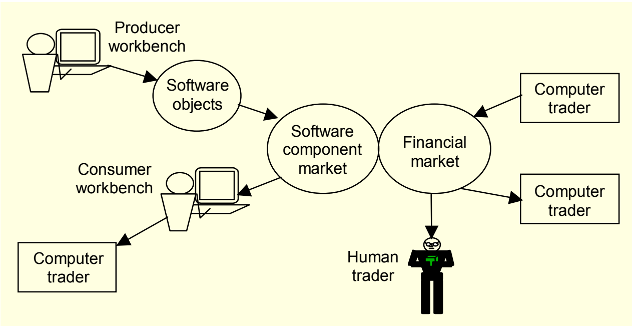


Fig. 1. Overview of SofTrade.

artificial market for the study of consumer behavior is its external validity. However, prior research has shown that consumers behave similarly in real and simulated environments, and so we can conclude that the results from the simulated settings are reasonably valid [24]-[26].

SofTrade consists of two markets, a *software component market* where consumers and producers interact, and a *financial market* where computerized programs interact. Starting from the upper left corner of the diagram in Fig. 1, the developers of components (*producers*) use the *Producer Workbench*[1)] to build software components, which are then placed in the software component market. The component buyers (*consumers*) search for and purchase components in this market. After components are purchased, consumers use the *Consumer Workbench* to assemble their components into more complex programs known as *Computer Traders*. These programs trade financial instruments (stocks) against each other in the financial market.

### 1. Producers and Consumers

A software component market requires agents that produce components (producers) and agents that purchase these components (consumers). In SofTrade, the role of component producer was filled by Master's degree students in computer science (CS). In the context of a course in object-oriented design and programming, these *technical* students were required to produce components for a group of *non-technical* students who served as the component consumers. Specifically, the consumers were MBA students. As part of a course called "Program Trading," the business students were required to search for and purchase the components created by the CS students. The business students were then required to assemble these components into computer traders without writing code by using the Consumer Workbench. The producers' grades were determined by how successful they were at selling components to consumers. Consumers were evaluated by subtracting the cost of their components from the profits made by their computer traders in the simulated financial market.

A total of twelve producer teams (two persons per team) and twenty individual consumers participated in this study. For both groups, 60% of the course grade depended on how well they fared in the simulation.

It should be emphasized that there was a strict division of expertise between the technical and non-technical student groups. This division of expertise created a natural incentive for the producers to want to understand the consumer's needs. Moreover, the division of skills in SofTrade mirrors the division of skills most often found in software production, where producers do not necessarily have extensive knowledge of the domain for which they are producing software code.

### 2. Software Component Market: Component Distribution Channel

The software component market was implemented on the World Wide Web as a set of web pages and scripts for finding, purchasing, and obtaining information about software components. In the market, producers offer their components at a fixed price, and any consumer can acquire any component by paying the appropriate amount. As a result, in the market, consumers are not guaranteed to get a unique component because other consumers can get a copy of the same component.

### 3. Financial Market: The Component Domain

The financial market in SofTrade is a double continuous auction similar to that used by Gode and Sunder [27]. This auction is a multilateral process in which buyers as well as sellers can enter orders (bids and asks). This particular mechanism was selected for two reasons. First, the choice is realistic because major stock, commodity, currency, and many other markets are organized as double auctions. Second, as demonstrated in Gode and Sunder's work [28], laboratory double auctions with human traders are known to yield data that approximates equilibrium predictions of economic theory.

## IV. Study of Perceived Subjective Features of Software Components and Consumer Behavior

### 1. Identifying Perceived Subjective Features of Software Components

#### *A. Generating Questions*

To identify the perceived subjective features affecting consumer behavior, we first conducted interviews with a group

1) 'Workbench' means a place for assembling components as a plate in Lego block.

of consumers who had participated in a preliminary study one year before. At that time, interviewees were asked what characteristics of software components they thought were important in developing their preferences and in making their purchasing decisions. Based on the subjective features cited in the interviews, we generated the 10 question items shown in Table 1.

### B. Gathering Data

After the questions were generated, the questionnaire was implemented on the Web. Then, consumers were asked to complete the questionnaire for all components in the market with a 5 point scale (1: Strongly Disagree, 5: Strongly Agree). Twenty consumers rated 23 components in the second market session and 26 components in the third. We provided consumers with a hard copy of the Web advertisements generated by the producers. In addition, they were able to browse the advertisements on the Web themselves.

### C. Data Analysis

Following the evaluation of software components by the consumers, a factor analysis on the average response of the 20 consumer raters was performed for each question. Although a data analyst might hesitate to use a multivariate technique like factor analysis with only 20 raters, this design actually offered a large number of data points (20 raters × 49 components × 10 items) since each rater rated every component. By averaging ratings across raters, it was possible to severely reduce the noise in the data, thus creating sufficient stability in the data for a multidimensional scaling analysis [29]. For the analysis, factors were first extracted with PROC FACTOR in SAS, using the principal factor analysis routine with varimax rotation.

### D. Results

The data analysis identified three factors, which were labeled as (perceived) functional richness, clarity, and flexibility. Table 1 shows how each question is clustered around each factor by using the highest factor loading for each question.

The Cronbach Alpha was 0.948 for the functional richness questions, 0.911 for the clarity questions, and 0.918 for the flexibility questions.

## 2. Developing Hypotheses between Perceived Subjective Features and Consumer Behavior

After identifying consumers' perceived subjective features of software components, the next goal was to link each of the three perceived component subjective features identified in the previous part of the study (clarity, functional richness, and flexibility) to consumer behavior—consumer preferences and purchasing behavior.

Table 1. Factor clustering.

| | Richness | Clarity | Flexibility |
|---|---|---|---|
| The algorithms in this component are complex. | X | | |
| This component can be used to build many different rules. | | | X |
| The information on the Web for this component is clear. | | X | |
| It is easy to customize this component. | | | X |
| It is difficult to understand the functionality of this component. | X | | |
| This component executes simple calculations. | X | | |
| I find the description of the component is believable. | | X | |
| It is difficult to combine this component with other components. | X | | |
| The examples for using this component are helpful. | | X | |
| This component is useful. | | | X |

### A. Perceived Clarity

Perceived clarity of a software component is the measure of how clearly the software component is represented to consumers. Since consumers in this environment collected product information mainly through the web advertisements, perceived clarity here indicates how clear and helpful consumers thought the web advertisements were for using a particular component.

Perceived clarity is expected to have a positive influence on consumer preferences and purchasing behavior. The importance of consumers' perceived clarity about software components can be understood in two ways: *ease of finding* and *ease of understanding*. In software component reuse research, representing reusable components so that they can be found and understood effectively has been thought to be one of the most critical factors in successful component reuse [7], [30]. The way in which software components are represented is important for reuse because, as has been noted in the past, "it affects the likelihood that a reusable part will be found and understood by a potential user" [7].

Many software reuse researchers have suggested that consumers are likely to prefer components that are easy to find because they make reuse easier and more economical by reducing search costs [24]. The effect of ease of understanding on users' acceptance of a system has likewise been emphasized by many researchers [31]. The rationale behind the importance of ease of understanding is rather simple: if users do not understand what the system can do for them, they cannot move even one step with it.

Previous research [31], [32] has stressed the importance of product documentation (which serves as a user aid and tutorial) in helping users find and understand a system effectively. In SofTrade, advertisements and help files on the producers' web pages serve as the main source of information about software components. If advertisements and help files can help consumers find and understand software components easily by providing them with required information, we can expect that this documentation will enhance consumer preferences and possibly induce consumers to purchase the product. Therefore, we hypothesize positive effects of perceived clarity on preferences and purchasing behavior.

**Hypothesis 1(a).** Perceived clarity of a software component would have a positive influence on consumer preferences.

**Hypothesis 1(b).** Perceived clarity of a software component would have a positive influence on consumers' purchasing behavior.

### *B. Perceived Functional Richness*

Perceived functional richness of a software component is defined as the level of functional robustness that the software component has. With software components, consumers combine different functional subjective features to accomplish tasks. For example, let us assume that a consumer needs an output "O" to do a task (see Fig. 2), and output O is based on three basic transformations: *x*, *y*, and *z*. If components $c_1$, $c_2$, and $c_3$ have the functions necessary to generate transformations *x*, *y*, and *z*, then the consumer should combine $c_1$, $c_2$, and $c_3$ to get output O. However, if component $c_4$ can generate output O by itself, then component $c_4$ will eliminate the effort and time that the consumer would expend to combine $c_1$, $c_2$, and $c_3$. In this example, component $c_4$ can be said to be more functionally rich than each of the other three components, and consumers can save time and effort by using $c_4$. In addition, by using $c_4$, consumers can also save the time and effort that would otherwise be required to find the appropriate simple components ($c_1$, $c_2$, and $c_3$ in this example) among the various components available to achieve the same output O.

Here, it was hypothesized that consumers would prefer and tend to purchase more functionally rich components because a functionally rich component can do more work for consumers with less time and effort [33].

**Hypothesis 2(a).** Functional richness would have a positive effect on consumer preferences.

**Hypothesis 2(b).** Functional richness would have a positive effect on consumers' purchasing behavior.

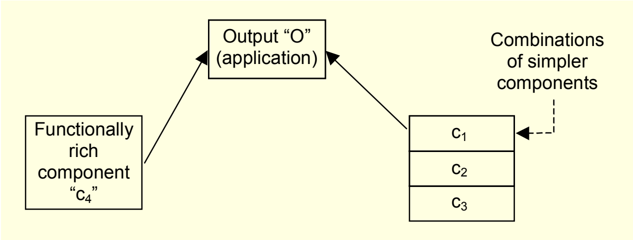


Fig. 2. Diagram showing potential benefit of using functionally rich component in combining components into an application.

### *C. Perceived Flexibility*

Perceived flexibility of a software component is how flexibly the component combines with other components. Perceived flexibility reflects the scope of applicability of a software component in building applications. As suggested by software reuse researchers [12], component flexibility (or generality) is so important because a component with broad applicability can greatly increase the total number of reuse opportunities. Likewise, perceived flexibility has been thought to have an important effect on system users' acceptance of a system or their subsequent performance in using the system [34]. Therefore, it is likely that components perceived as flexible would have a stronger positive impact on consumer behavior than components perceived as inflexible.

**Hypothesis 3(a).** Perceived flexibility of a software component would have a positive influence on consumer preferences.

**Hypothesis 3(b).** Perceived flexibility would have a positive effect on purchasing behavior.

### *D. Consumer Preferences*

It was expected that consumers' preferences would play an important role in this consumer decision model because, all other things being equal, consumers would like to buy what they like. Traditionally, affective attitudes (preferences) are believed to play a pivotal role in explaining consumer behavior—especially purchasing behavior [15].

**Hypothesis 4.** Consumer preferences with respect to using software components would have a significant positive effect on purchasing behavior.

### *E. Component Prices and Discounts*

Given that the scope of our study is expanded to a market environment where users choose among multiple software components from different vendors, situational constraints—price and discount—are reflected as influencing purchasing behavior in the model [17]. In this study, producers (vendors) could freely decide the prices of their components and introduce price discounts to consumers. Similar to other goods

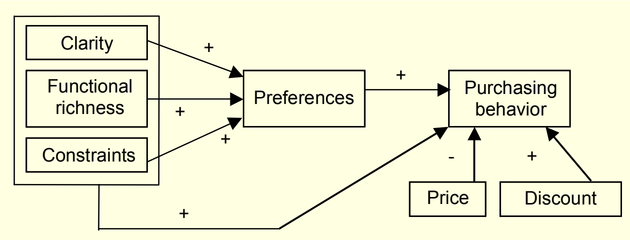


Fig. 3. Consumer behavior model based on consumers' perceived subjective features on software components.

and services in the market, the cost of using software components is a very important consideration for potential users in their adoption decision. Indeed, price is considered a pivotal determinant of consumer behavior [35].

Prices of software components are expected to have a negative impact on purchasing behavior, because, all other things being equal, people generally would like to buy things at a cheaper price. By the same token, discounts on software components are expected to have a positive influence on purchasing behavior.

**Hypothesis 5.** Price would have a significant negative effect on purchasing behavior.

**Hypothesis 6.** Price discount would have a positive effect on purchasing behavior.

Based on the discussion of the various factors covered in this section that are likely to affect consumer behavior—perceived clarity, perceived functional richness, perceived flexibility, consumer preferences, and component prices and discounts—a model was established to test these hypotheses (see Fig. 3).

## V. Experimental Method

### 1. Component

All components were developed in C++ language. Each *component* was able to contain multiple methods and each *method* dictated a distinctive function, which allowed it to operate with other methods. Thus, consumers were able to combine methods associated with different components and could combine methods from the same component to build trading strategies.

To control the size of a component, a guideline of a maximum of ten methods per component was recommended to producers. Producers were asked to place a maximum of four components at any one time to control the total number of transactions in the market.

### 2. Procedure

Each simulation cycle consisted of a software component market session and a financial trading market session. Three simulation cycles were conducted during a five-week period. Each software component market session opened on a Friday, with a two-week break between market sessions two and three. Each financial trading market took place on the Wednesday following the opening of a software component market session. This schedule imposed deadlines by which both producers and consumers had to build their components (producers) or computer traders (consumers). In SofTrade, components are actually *rented* for a week. After they are used once, they are returned to the producers. The producers may, and often do, modify their components and advertising after receiving feedback from the consumers; at this point they may generate a new line of components.

#### *A. Software Component Market Sessions*

Prior to each software component market session, producers advertised their components on the Web. Producers were free to set prices, develop promotional campaigns, and design advertisements. During each market session, consumers bought components from the producers' sites on the Web.

#### *B. Financial Trading Sessions*

Prior to Wednesday's financial trading session, consumers had several days to build their computer traders, using the components they had bought during the previous software component market session. During the financial trading session, consumers loaded their computer traders into the machines and let them trade with each other. During a single financial trading session, the market opened and closed several times (between 6 and 8 times). Consumers could observe how their computer traders were performing against other computer traders every time the market closed during the session. After each financial trading session, consumers were able to retrieve data about the behavior and performance of their own computer traders and those of the other students in the course.

### 3. Data Collection

Data was collected during cycles two and three. Data from the first cycle was eliminated because both producers and consumers were learning how to play their roles.

#### *A. Consumer Preferences*

Consumers were asked to sort all of the components on the market (23 components from the second market session and 26 components from the third session) in order of preference. In each sorting session, each participant received descriptions of the components in the software component market and was asked to sort them into six piles ordered according to preferences.

Use of this *monadic-rating* technique revealed each consumer's ranking of the components and the distance between them. This technique is especially useful when evaluating a large number of objects [18].

Participants were instructed to disregard the price of components in determining their preferences. This is because, as has been suggested by past research [17], price was thought to be a situational constraint that might affect choice behavior, rather than preferences.

Consumer preference for each component was calculated by averaging the scores for each component across the twenty consumers.

### *B. Factor Scores of Perceived Subjective Features*

Using the SCORE procedure in SAS, factor scores for each perceived component feature were calculated. These factor scores for each perceived feature comprised the data used in predicting consumer preferences and purchasing behavior in this study.

Factor scores are essentially regression estimates of the score for each component on each factor. Since the factors were uncorrelated, the factor score estimates were expected to be uncorrelated as well, as they essentially were (the correlations of the 3 factors were 0.0463, 0.0584, and -0.1002). This procedure produced a correctly specified regression [36] and made better use of the data than more traditional composite scoring, in which one takes the mean of the high loading items in the factor analysis [36].

### *C. Purchasing Behavior*

This information was automatically collected from the online market. The purchasing behavior measure was dichotomous. For each consumer in each purchasing session, components purchased were described as 1, otherwise 0.

### *D. Price and Discount*

Price refers to the price for each component in each session. Discount refers to the price discount for each component. This discount measure was dichotomous. For each consumer in each session, components under a discount plan were described as 1, otherwise 0. The reason for using a dichotomous discount variable was that the discount rate provided by almost all of the producers in the market was 15%.

## VI. Results

This section reports two sets of results: first, linear regression results of consumers' preferences on perceived component subjective features; and second, logistic regression results of consumer purchasing behavior on perceived subjective features, consumer preferences, price and discount.

Table 2. Regression analysis of consumer preferences on perceived subjective features.

| Variables | Coefficients | S.E. | p-value |
|---|---|---|---|
| Clarity | 0.0031 | 0.101 | 0.975 |
| Functional richness | -0.1230 | 0.109 | 0.265 |
| Flexibility | 0.7490 | 0.091 | 0.001 |
| Price | 0.0003 | 0.006 | 0.954 |

N=49, Adjusted $R^2$: 0.592, F-Statistic: 15.42

### 1. Linking Perceived Component Subjective Features and Consumers' Preferences

To determine the relationship between perceived component subjective features and consumer preferences, the linear regression method was used. To verify whether the participants in the component sorting followed the instructions to disregard price, the price of each component was also entered into the model.

The results (see Table 2) indicate that flexibility was the only significant variable explaining consumer preferences. The insignificant effect of the component price verifies that the price factor did not have an impact on the preference rating as expected. The results also showed that perceived flexibility by itself explained about 60% of the variance in consumer preferences.

### 2. Predicting Purchasing Behavior Using Perceived Subjective Features

To predict consumer purchasing behavior, the logistic regression method was used. The method entails regressing purchasing behavior (1/0) of each consumer for each component on perceived subjective features, consumer preferences, price, and discount.

Table 3 shows the results of the logistic regression model. Among the three perceived subjective features, functional richness was found to be very significant, and clarity was found to be marginally significant. Note that flexibility was not found to be significant, even though it was the only significant factor in the relationship between consumer preferences and perceived subjective features. Consumer preferences did not predict purchasing behavior well. Price and discount were both significant, as expected. The model chi-square was significant ($p<0.001$), indicating that the model explained the relationships well [37]. Figure 4 summarizes the results graphically.

Table 3. Effects of explanatory variables on purchasing behavior.

| Variables | Coefficients | S.E. | p-value |
|---|---|---|---|
| Preferences | 0.1999 | 0.1812 | 0.270 |
| Clarity | 0.2437 | 0.1399 | 0.081 |
| Functional richness | 0.4551 | 0.1436 | 0.001 |
| Flexibility | 0.0468 | 0.1693 | 0.782 |
| Price | -0.0202 | 0.0083 | 0.015 |
| Discount | 0.6676 | 0.3316 | 0.044 |

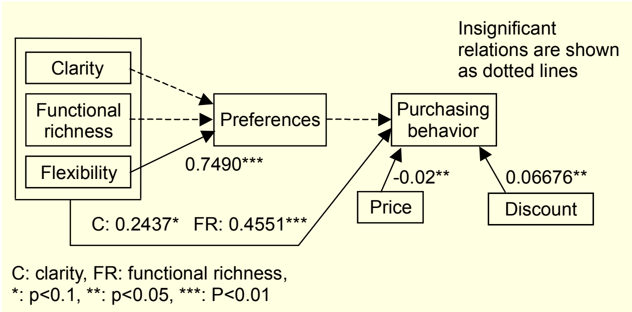


Fig. 4. Graphic presentation of results in Table 3.

## VII. Discussion

Overall, the results indicate that perceived subjective features of the software components investigated in this study can be important in explaining consumer behavior in a software component market. Our findings extend those of Hong and Lerch [14] regarding the impact of objective features of software on consumers' preference and purchasing behavior. One important and interesting result was that flexibility was the only significant factor among the three perceived subjective features in explaining consumer preferences, while clarity and functional richness were the only subjective features significant in predicting purchasing behavior. This suggests that consumers' evaluation perspectives on software components can vary depending on the type of decision task they face—in this case, preference rating and purchasing decision. This is related to Payne, Bettman, and Johnson's view [38] that evaluations of alternatives can be affected by minor changes in the task environment. In other words, people adapt to situational changes such as time pressure. In our study, whether or not consumers have monetary and time costs associated with performing the decision task can be important factors that affect how consumers reach their preference and purchasing decisions. At the purchasing decision stage, consumers are implicitly committed to building an application within a certain time limit (that is, before the financial trading market). Additionally, consumers' purchasing decisions involve monetary costs that will be directly related to their performance[2]. The commitment to building an application using the software components in the market and the monetary costs incurred in the purchasing decision can motivate consumers to carefully evaluate the relative benefits and costs of using software components to build an application because their purchasing decision is critical to their performance and irreversible. In contrast, when consumers were asked to simply rate their preferences about using software components, they were under no imminent pressure to build an application using the software components. Moreover, no monetary cost was involved in simple preference rating. Indeed, for consumers, simply rating their affective feeling about software components was not directly related to their task performance. Without such cost and pressure associated with performing preference rating, consumers were likely to have been less motivated to consider all the benefit and cost aspects of using software components.

The elaboration likelihood model of Petty and Cacioppo [40] is useful in elucidating the relationships between perceived subjective features and consumer behavior at the preference rating stage. According to this view, the level of elaboration in a communication situation is a function of the motivation of the communication recipients. As mentioned earlier, the help files and advertisements provided by the producers were the main source of information that consumers could use to evaluate the software components. When consumers simply rated their preferences among software components and, thus were less motivated to scrutinize the communication arguments (help files and advertisements), perceived clarity—essentially the quality of documentation and advertisements—might not have been an important concern for them. Similarly, the low level of elaboration in the information on the software components could not allow consumers to properly understand the relative benefits of functionally rich components. Even if consumers could recognize the benefits of those components (such as saving time and effort), they might not be very concerned about saving time or effort because they were under no imminent pressure to build an application.

On the other hand, perceived flexibility had a significant impact on preferences. At the time of rating, consumers may have lacked a concrete, specific idea of the application they would build, and so did not know what components they would need. Bettman and Sujan [41] propose that, without concrete, specific criteria with which to evaluate alternatives, people are apt to use general criteria. From this perspective, it is understandable that the software components with general applicability (flexibility) may have seemed especially appealing to consumers, because they did not have any specific

2) Note that consumers' performance was calculated by subtracting the cost of their components from the profits made by their computer traders in the simulated financial market.

idea of the strategy, or had too many vague ideas for which flexible components might be used.

At the purchasing stage, however, the benefits of saving time and effort in building applications may have appealed to consumers who had become concerned with time and cost. Considering the critical effect of purchasing decisions on performance, consumers were likely to pay careful attention to the documentation and advertisements. The higher the clarity of the information is, the better it helps consumers easily find what they want and understand what the software components can do for them, thus helping reduce search time and shorten the learning curve. All other things being equal, consumers would like to buy components that minimize these costs. Moreover, consumers would not like to risk their money buying components that they find difficult to understand. Given these factors, it was not surprising that perceived clarity had a significant impact on purchasing behavior in this study.

The significant impact of functional richness on purchasing behavior also implies that consumers chose the benefits of saving time and effort in building applications. Keinan [42] suggested that people tend to respond to time pressure by narrowing the range of actions considered to reach a final goal state. Given the time limitation, consumers seemed to choose functionally rich components that would reduce the number of assembly steps. Another possible explanation for the significance of functional richness may be found in the nature of functionally rich components. Functionally rich components might provide consumers with a shorter cognitive distance to the application that they intend to implement. According to Krueger [16], cognitive distance is the amount of intellectual effort that must be expended by application developers to transform a software system from one stage of development to another. Small, basic components bear very little resemblance to the intended application. Therefore, with small, basic components, consumers need to go through a long transformation process to the intended application because these components require many transformations. The long transformation process, that is, the long cognitive distance, may tire consumers cognitively, thus causing errors and confusion when they visualize a programmed strategy. By contrast, with functionally rich components, consumers may be able to reduce the number of transformations required to reach the intended application. In this way, building an application with functionally rich components may be cognitively less demanding and less error prone for consumers. For consumers who were committed to building applications within a deadline, functionally rich components may have seemed more appropriate.

The limited explanatory power of perceived flexibility in predicting purchasing behavior may be understood with a similar line of reasoning. The benefit of using flexible components is that it enables consumers to try to include various ideas and conditions into trading programs by assembling them in many different ways. However, at the purchasing stage, where the costs (in both time and cognitive effort) of testing various ideas in applications had become a serious matter, flexibility may not have seemed especially appealing to consumers.

The results imply that software component consumers may be myopic in evaluating the benefits of software components. In this study, consumers shifted their evaluative perspectives as the situation changed. That is, when consumers were committed to developing an application within a limited time, they chose the benefits of functional richness while overlooking the benefits of flexibility. Indeed, the importance of component flexibility lies in the component's broad applicability that can greatly increase the total number of reuse opportunities. Thus, flexible components that can be reused for various applications may save a lot of time, effort, and money for users in the long run. Despite the long-term value of flexible components, consumers seemed to focus on short-term benefits that could save them time and effort in accomplishing their imminent task. This is similar to the view of Davis and others [19] that even if a consumer does not like a system, the expectation of improving his/her performance by choosing it may lead the consumer to purchase it. Although more empirical research is necessary on this aspect based on real field data, assuming that a component-based application project in a *real work setting* is to be conducted within a certain deadline as most software application projects are, it may be possible to observe component users choose functional richness over flexibility to improve their short-term performance. This further implies that software component producers should carefully design components so that the components can reasonably satisfy both aspects most of the time.

In summary, the results of this study suggest that the relationships between perceived subjective features of software components and consumer behavior are dynamic and complex. The results also suggest that by better understanding the potential impact of each perceived feature on consumer behavior, producers will be able to develop successful software components that are more attractive to consumers, and thereby increase the reuse of software components.

## References

[1] M.D. McIlroy, "Mass Produced Software Components," P. Naur and B. Randell (eds.), *Software Engineering; Report on a Conference by the NATO Science Committee,* Brussels, Belgium, NATO Scientific Affairs Division, 1968, pp. 138-150.

[2] A. Lynex and P.J. Layzell, "Organisational Considerations for Software Reuse," *Annals of Software Engineering*, vol. 5, 1998,

pp. 105-124.
[3] C. Szyperski, *Component Software: Beyond Object-Oriented Programming,* New York, NY, ACM Press, 1997.
[4] J.M. Favaro, K.R. Favaro, and P.F. Favaro, "Value Based Software Reuse Investment," *Annals of Software Engineering*, vol. 5, 1998, pp. 5-52.
[5] WB. Frakes, "Software Reuse Empirical Studies," W. Schafer, R. Prieto-Diaz, and M. Matsumoto (eds.), *Software Reusability*, Chichester, UK, Ellis Horwood, 1994, pp. 153-160.
[6] R.L. Glass, "Reuse: What's Wrong with This Picture?" *IEEE Software*, vol. 15, no. 2, 1998, pp. 57-59.
[7] W.B. Frakes and P.B. Gandel, "Representing Reusable Software," *Information and Software Technology*, vol. 32, 1990, pp. 653-664.
[8] J.S. Poulin, "Reuse: Been There, Done That," *Communications of the ACM*, vol. 42, no. 5, 1999, pp. 98-100.
[9] CW. Church, "Technology Issues of Software Reuse," M. Sarshar (ed.), *Systematic Reuse: Issues in Initiating and Improving a Reuse Program*, London, U.K.: Springer-Verlag. 1996, pp. 64-69
[10] D. Judicibus, "Reuse: A Cultural Change," M. Sarshar (ed.), *Systematic Reuse: Issues in Initiating and Improving a Reuse Program,* 1996, London, U.K.: Springer-Verlag, pp. 44-51.
[11] D.J. Reifer, *Practical Software Reuse*, Chichester, NY, Wiley, 1997.
[12] B.H. Barnes and T.B. Bollinger, "Making Reuse Cost-Effective," *IEEE Software*, vol. 8, no. 1, 1991, pp. 13-24.
[13] W.B. Frakes and C.J. Fox, "Quality Improvement Using a Software Reuse Failure Modes Model," *IEEE Tran. Software Engineering*, vol. 22, 1996, pp. 274-279.
[14] S.J. Hong and F.J. Lerch, "A Laboratory Study of Consumers' Preferences and Purchasing Behavior with Regards to Software Components," *ACM SIGMIS Database*, vol. 33, no. 3, 2002, pp. 23-37.
[15] J.F. Engel, R.D. Blackwell, and P.W. Miniard, *Consumer Behavior*, 8th ed., Fort Worth, TX, Dryden Press, 1995.
[16] C.W. Krueger, "Software Reuse," *ACM Computing Surveys*, vol. 24, 1992, pp. 131-183.
[17] J.R. Hauser and P. Simmie, "Profit Maximizing Perceptual Positions: An Integrated Theory for the Selection of Product Features and Price," *Management Science*, vol. 27, 1981, pp. 33-56.
[18] P. Kotler, *Marketing Management*, 9th ed., Upper Saddle River, NJ, Prentice Hall, 1997.
[19] F.D. Davis, R.P. Bagozzi, and P.R. Warshaw, "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," *Management Science*, vol. 35, 1989, pp. 982-1003.
[20] G. Moore and I. Benbasat, "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," *Information Systems Research*, vol. 2, 1991, pp. 192-222.
[21] C.R. Plouffe, J.S. Hulland, and M. Vandenbosch, "Research Report: Richness Versus Parsimony in Modeling Technology Adoption Decisions: Understanding Merchant Adoption of a Smart Card-Based Payment System," *Information Systems Research*, vol. 12, 2001, pp. 208-222.
[22] V. Venkatesh, "Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion into the Technology Acceptance Model," *Information Systems Research*, vol. 11, 2000, pp. 342-365.
[23] S. Taylor and P.A. Todd, "Understanding Information Technology Usage: A Test of Competing Models," *Information Systems Research*, vol. 6, 1995, pp. 144-176.
[24] R.R. Burke et al., "Comparing Dynamic Consumer Choice in Real and Computer-Simulated Environments," *J. Consumer Research*, vol. 19, 1992, pp. 71-82.
[25] A. Gabor, C.W.J. Granger, and A.P. Sowter, "Real and Hypothetical Shop Situations in Market Research," *J. Marketing Research*, vol. 7, 1970, pp. 355-359.
[26] J.R. Nevin, "Laboratory Experiments for Establishing Consumer Demand: A Validation Study," *J. Marketing Research*, vol. 11, 1974, pp. 261-268.
[27] D.K. Gode and S. Sunder, "Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality," *J. Political Economy*, vol. 101, 1993, pp. 119-137.
[28] D.K. Gode and S. Sunder, "Lower Bounds for Efficiency of Surplus Extraction in Double Auctions," D. Friedman J. Rust (eds.), *The Double Auction Market*, Reading, MA, Addison-Wesley, 1993, pp. 199-219.
[29] L. Jones and L. Koehly, "Multidimensional Scaling," G. Keren, and C. Lewis (eds.), *A Handbook for Data Analysis in the Behavioral Sciences*, Hillsdale, NJ, Lawrence Erlbaum, 1993, pp. 95-163.
[30] W.B. Frakes and S. Isoda, "Success Factors of Systematic Reuse," *IEEE Software*, vol. 11, no. 5, 1994, pp. 14-19.
[31] R. Nickerson, "Why Interactive Computer Systems Are Sometimes Not Used by People Who Might Benefit from Them," *Int'l J. Man-Machine Studies*, vol. 15, 1981, pp. 469-483.
[32] A.L. Paxton and E.J. Turner, "The Application of Human Factors to the Need of the Novice User," *Int'l J. Man-Machine Studies*, vol. 20, 1984, pp. 137-156.
[33] J.W. Payne, "Contingent Decision Behavior," *Psychological Bulletin*, vol. 92, 1982, pp. 382-402.
[34] N.C. Goodwin, "Functionality and Usability," *Communications of the ACM*, vol. 30, 1987, pp. 229-233.
[35] K.B. Monroe, *Pricing: Making Profitable Decisions*, 2nd ed., New York, NY, McGraw-Hill, 1990.
[36] K. Bollen and R. Lennox, "Conventional Wisdom on Measurement: A Structural Equation Perspective," *Psychological Bulletin*, vol. 110, 1991, pp. 305-314.
[37] R.J. Rummel, *Applied Factor Analysis*, Evanston, IL, Northwestern University Press, 1970.

[38] S. Menard, "*Applied Logistic Regression Analysis*," Thousand Oaks, CA, Sage Publications, 1995.
[39] J.W. Payne, J.R. Bettman, and E. Johnson, "*The Adaptive Decision Maker*," New York, NY, Cambridge University Press, 1993.
[40] R.E. Petty and J.T. Cacioppo, "The Elaboration Likelihood Model of Persuasion," L. Berkowitz (ed.), *Advances in Experimental Social Psychology*, New York, NY, Academic Press, vol. 19, 1986, pp. 123-205.
[41] J.R. Bettman and M. Sujan, "Effects of Framing on Evaluations of Comparable and Non-comparable Alternatives by Expert and Novice Consumers," *J. Consumer Research*, vol. 14, 1987, pp. 141-154.
[42] G. Keinan, "Decision Making under Stress: Scanning of Alternatives under Controllable and Uncontrollable Threats," *J. Personality and Social Psychology*, vol. 52, 1987, pp. 639-644.

**Janghyuk Lee** is an assistant professor of marketing with Korea University Business School (KUBS). He received the BA in international economics from Seoul National University, and the MS and PhD from ESSEC, France. He worked for the University of Reading, UK, and HEC School of Management before joining KUBS. His research interests include social network analysis (spread of online word of mouth), Internet advertising (optimal display ad scheduling), and customer equity management (customer lifetime value). He has published in *Marketing Science*, *Journal for Services Marketing*, and *Communications of the Association for Information Systems*.

**Se-Joon Hong** is an associate professor of management information systems with the Business School of Korea University. He obtained his PhD in information systems from the Graduate School of Industrial Administration (GSIA) at Carnegie Mellon University. His current research interests are ubiquitous computing, IT adoption and diffusion, and human factors in information systems. He has papers published or forthcoming in *Information Systems Research*, *Management Science*, *International Journal of Human-Computer Studies*, *Communications of the ACM, IEEE Transactions on Engineering Management, Decision Support Systems*, *International Journal of Electronic Commerce*, *The DATABASE for Advances in Information Systems*, *Electronic Markets,* and others.

**Yeong-Wha Sawng** is a senior researcher of the Service Strategy Research Team with the Electronics and Telecommunications Research Institute (ETRI), Korea. He received the BA and MIM in Management from Whitworth University, USA, and the PhD in Technology Management from Hanyang University, Seoul, Korea. He joined ETRI in 2000, and has been working in the areas of digital convergence, IT policy, mobile application, technology management, and business strategy. His research interests also include high-tech marketing, technology management strategy, e/m-Biz models, and consumer behavior. His technical papers have been published in various international journals, including *International Journal of Technology Management*, *Journal of Research and Practice in Information Technology*, *International Journal of Satellite Communication and Networking*, and *International Journal of Innovation and Technology Management*.

**Ju Seong Kim** received the MS degree in business administration from Seoul National University, Seoul, Korea, in 1983 and PhD degree in finance from Seoul National University in 1996, respectively. He joined Electronic and Telecommunications Research Institute (ETRI) in 1985 and since 1999 he has been working as a principal researcher. His research interests include IT industry analysis and policy, economic analysis of R&D project, innovation strategy, and competitiveness analysis of network equipment industry. His papers have been published in various international and Korean journals including *Small Business Economics* and *Korean Management Review*. He is now an editor of *ETRI Journal*.