# Transaction Processing Systems

**Sasan Rahmatian**
*California State University, Fresno*



## GLOSSARY

**ACID properties** A transaction is characterized by four properties referred to as the ACID properties: atomicity, consistency, isolation, and durability.
**database update** The real effect of a transaction; it takes one of three forms: add, change, and delete.
**logical conception** A view of a TPS that focuses on *what* information flows into/out of it, and *what* processes are performed on that information. It is relatively stable and permanent.
**master file** A file that contains data about permanent entities, such as customers, products, suppliers, and employees. Its contents are updated by a transaction file.
**physical conception** A view of a TPS that focuses on the technology used in entering the data, producing the information, and performing the processes. It is dynamic and transient due to frequent technological advances.
**ripple effect** Occurs when transaction data travel to all the other parts of the organization where they are needed.
**scalability** The ability of an application or platform to be expanded in terms of capacity to accommodate a larger number of users or transactions than originally planned without requiring significant changes in procedure.
**transaction** A business event whose relevant attributes need to be recorded internally (in the corporate database) as well as externally (for the benefit of external stakeholders, such as customers, suppliers, business partners, and regulators) due to the impact this information will later have on other operations of the organization.
**transaction file** A file that contains data about a particular class of transactions, such as sales, reservations, returns.

**A TRANSACTION PROCESSING SYSTEM (TPS)** is a system that captures, enters, stores, retrieves, and processes the relevant details of business events and generates the information and documents necessary for running the business. As such, it is a subset of the operational subsystem of the organization, recording the work done. The data captured and stored by a TPS serve two purposes. First, to support day-to-day, routine operations by being made accessible to those parts of the organization (as well as to external entities) where they are needed. Second, to feed the management reporting system and produce performance reports about the effectiveness and efficiency of the operations. The usefulness of transaction data goes beyond mere operations because large volumes of detailed data can be used as a historical reference for forecasting, identifying trends, and measuring performance. Understanding a TPS can be accomplished along two different dimensions: externally versus internally, and logically (what it does) versus physically (the technology based on which it works). Although a TPS is primarily concerned with day-to-day operations, it does play a critical strategic role in the organization, for it is with the operational level of the organization that customers and suppliers interact, and if the TPS is not working satisfactorily, then those external relations tend to suffer.

## I. THE LARGER FRAMEWORK

Any organization—regardless of size, industry, and profit orientation—exists to serve its clientele by adding value to the resources obtained from its external environment. The operational level of the organization consists of those activities that perform the value-adding work on a day-to-day basis. The management control level consists of those activities that compare the results accomplished with the goals set by management in order to identify and correct any possible variances between the two. The data captured and stored by a TPS serve two purposes. First, to support day-to-day, routine operations by being made accessible to those parts of the organization (as well as to external stakeholders) where they are needed. Second, to feed the management reporting system and produce performance reports about the effectiveness and efficiency of the operations. Regardless of which purpose is served, a TPS stores the data it captures in a database for later use. As such, the database is a buffer (1) between the TPS and the management reporting system, and (2) between the TPS at one time and the TPS at a later time (Fig. 1).

## II. TRANSACTION DEFINED

*Trans* means "beyond," as in *transatlantic.* A transaction is the action that an entity performs beyond itself, i.e., on another entity. Hence a transaction, at the simplest level, can be defined as an event involving the action of one entity on another. The word *exchange* is often used to characterize a transaction.

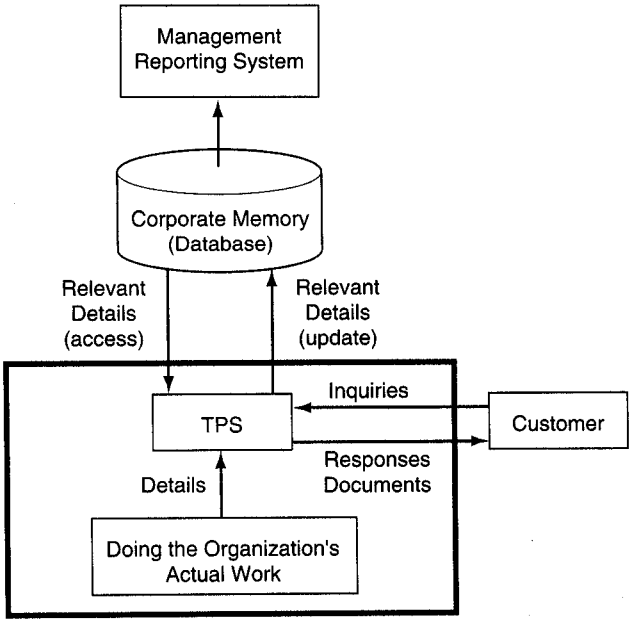


**Figure 1** The larger framework of TPS.

In earlier, simpler times, when business operations were run on a small scale, it was not necessary to use information technology to get accurate status information about business processes. No technology was needed to mediate between the users and the physical reality they were trying to measure or manipulate. For instance, when a customer walked into a store and asked about the availability of a certain item, it would be possible for the clerk to go to the storage room, look up availability, and then report that information back to the customer in a matter of seconds. As this organization grew larger and larger, the size of its warehouse increased accordingly. Beyond some critical threshold point, it would take too long to check the item availability manually.

This is the point at which technology comes in to mediate between humans and the reality they are trying to control (Fig. 2). In a computer-mediated environment, the relevant details of all the sales and shipments received are entered into a computer system. The computer system then automatically updates quantity on hand of any item sold or purchased. Hence, the old scenario would be replayed as follows: When a customer walks into the store and asks about the availability of a certain item, the clerk has to look up its availability on the computer. In other words, the computer has now become the clerk's window into the real world. The picture of the world provided by the system is the only practical way of knowing what is happening in the world. What the system says is real—and only that—can be considered real.

From the above picture emerges the importance of capturing and recording events that impact other parts of the organization, and thus need to be retrieved at a later time. But an event has many different attributes. Only those attributes relevant to later retrieval and use need to be captured and stored. This is the basis on which the concept of transaction is defined as *a business event whose relevant attributes need to be recorded internally (in the corporate database) as well as externally (for a customer, a supplier, a business partner, etc.) due to the impact this information will later have on other operations of the organization and its stakeholders.* Being an event, a transaction takes place in time. It always has a date and time stamped on it. This is significant for reporting purposes, because management can later measure performance down to any unit of time desired ("How many Big Macs are sold per hour?").

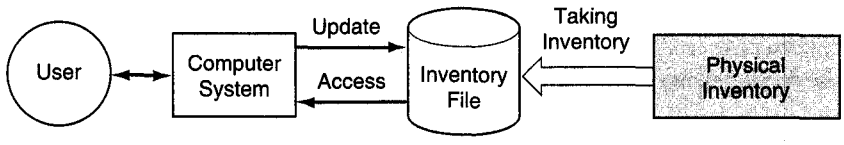


**Figure 2** The computer system mediating between the humans and physical reality in an inventory application.

Updates brought about by a transaction are of three types:

- Adding a new record, such as when a new customer signs up
- Changing an existing record, such as changes in credit available for a customer when that customer makes new purchases or returns already purchased products
- Deleting a record, such as when a product is discontinued

Thus defined, transactions are an ancient concept, as old as the beginning of writing. In the Fertile Crescent, archeologists have found clay tablets written with cuneiform symbols that record commercial transactions such as sale of slaves. Some of these tablets are thousands of years old.

In a world in which the virtual is regarded as the image of the physical, what is beginning to matter more and more is not what actually happened, but what the system recorded as having happened. In Fig. 3, four scenarios can be distinguished:

1. *An event happened and it was recorded.* This is a case in which the virtual accurately reflects the physical.
2. *An event happened but it was never recorded.* Consider a customer complaint that was never recorded, or was recorded and later erased (intentionally or otherwise). Because this event was not recorded, it would not leave a trace on the organization's collective memory (i.e., the database), hence failing to set in motion the actions that a customer complaint is supposed to trigger, such as a letter of apology to the customer and an investigation into the internal processes that were responsible for the complaint (defective product, discourteous employee, etc.). Unless the customer complained again, this information would not be acted on because of the organization not remembering it.
3. *An event that never happened was mistakenly recorded as having occurred.* This could take place due to fraud or forgery, but more often it may be due to incorrect processing. Hence, items may actually exist in stock but are shown by the computer system not to be there. Thus a salesperson, checking inventory on-line, could mistakenly see an item as not available and thus forego a sale while the item actually exists in inventory.
4. *An event that never happened was not recorded.* Again, in this case, the virtual accurately reflects the physical.

In short, in the TPS mind-set, if it was not recorded, it did not happen; if it was, it did.

Three important points need to be made about the concept of transaction. One subtlety associated with the concept of event is that *sometimes certain occurrences*

| | It was recorded | It was not recorded |
|---|---|---|
| **It really happened** | **1. OK**<br>Example:<br>A rental item is returned on time and is recorded as having been returned on time | **2. It did "not happen"**<br>Example:<br>A rental item is returned on time but is mistakenly stamped as having been returned late. |
| **It really did not happen** | **3. It "happened"**<br>Example:<br>A rental item is returned late but is mistakenly stamped as having been returned on time. | **4. OK**<br>Example:<br>A rental item is returned late and it is stamped as having been returned late. |

**Figure 3** What happened versus what was recorded. The event: Returning a rental item (such as a videotape) on time.

*that do not happen but are expected to happen constitute events.* For instance, a customer's failure to make a payment on his credit card against an assigned deadline may appear as a nonevent. Yet, because of the expectation of the receipt of the payment by the deadline, that nonevent will become an event, triggering finance charges on the next account cycle.

The second subtlety is associated with the concept of relevance. Transactions record the relevant attributes of events, but relevant to what? Two types of relevance can be distinguished: operational and managerial.

The *operational relevance* of an event is derived from other operations in the organization that use the data captured about this event. The simplest way to bring out the operational relevance of an attribute of an event is to ask. "What operations in the organization will later use this piece of information, and how?" For instance, consider the business event in which a salesperson takes a prospective customer to lunch. The amount spent would be relevant because it will be used to later reimburse that salesperson.

The *managerial relevance* of an event is derived from controls set by management to assure that things go according to plan. Returning to the above example, the amount spent on lunch would also have a managerial relevance in that it should not exceed an upper limit preset by management. But more to the point would be the managerial relevance of the *purpose* of the lunch. If that piece of data is not captured, how will management be able to ascertain whether that expenditure was worth it?

The managerial type of relevance can be extended beyond control to include strategic objectives. It used to be that as soon as a customer walked through the door of the corner shop, the proprietor knew who he was and what he wanted. With the scale of business operations increasing immensely over time, organizations have become more impersonal. Strategic conceptions of TPS aim at bringing those days back. Retailers are trying to bring back the "personal touch" using technology to compensate for impersonal contact. Regardless of the scale of an organization's operations, it is good business to know who the organization's customers are and what they want. The problem for large organizations is that they do not even know who their customers are. A large supermarket might know how many units of which item it sold when, but it usually knows much less about the people who buy them. A strategically conceived TPS allows for *micromarketing*—the ability to identify small groups of customers, even individuals, and sell them a product they need, or motivate them to buy such a product by sending them a customized coupon. In terms of customer recognition, this is possible through special-privilege cards that entitle the customer to discounts on selected items. When the customer swipes the card in the machine, his name comes up on the monitor, and he is greeted by the supermarket staff by name.

The third subtlety has to do with the notion of update versus access. Accessing a database to perform a query or produce a report does not ordinarily constitute a transaction because it does not involve an update to the database. However, *a query is considered a transaction if the query itself is meant to be tracked.* For instance, when a customer inquires into the availability of an item, this would ordinarily count as a read/access operation. However, management may at some point realize that the inquiries themselves are a way of tracking demand. At that point, each inquiry would be tracked in terms of the items requested and their availability. A high level of unavailability of an item would signal management to order or produce a higher level of it.

## III. TPS DEFINED

Having defined transaction, we can now proceed to define the term TPS. TPS is a system that captures, enters, stores, retrieves, and processes the relevant details of business events, and generates the information/documents necessary for running the organization and interfacing with external entities, such as customers. The *processing* part of the TPS has to do with the various activities involved in running it, such as the following:

- Capturing the data as close to the source as possible
- Entering them into the system in a manner as effectively and efficiently as possible
- Storing them in the database
- Retrieving them from the database for further processing
- Transforming them from the raw form to information useful to the intended user.

The *systems* part of the TPS has to do with planning and designing the above elements so that they work in sync to produce the organizational objectives behind the TPS. The *transaction* part of the TPS is somewhat more complicated. TPSs are characterized by

- Large amounts of input/output
- Large number of users
- Huge storage requirements
- Low computational complexity

- Fast input/output as well as processing capabilities
- A high degree of concern for potential security-related problems
- A high degree of concern for reliability and fault-tolerance.

A TPS is the lifeblood of an organization. For any business-related event occurring in an organization, a number of design decisions need to be made, such as these:

1. Is the event worth recording? If so, why? Customers calling to ask questions about their bill—is this event worth capturing? Yes, if (among other reasons) the organization is trying to find the most common reasons for customer calls, in order to automate responses to the most frequently asked questions.
2. If the event is worth recording, what attributes need to be captured, and why? Is the identity of the customer service representative (CSR) handling the customer's question in the above example worth documenting? Yes, if (among other reasons) the organization is trying to track customer satisfaction and train those CSRs who need more training.
3. In what form(s) should the event be documented? When a customer returns a product, there is a need to record this event not only internally (the database) but also externally (a credit memo issued to the customer).
4. How should the event be recorded? It used to be that prices were either stamped or tagged on products in supermarkets. The checker would then manually key this price into the cash register. This was both slow and prone to errors. With the advent of bar codes, this event was automated, thus making data capture faster as well as more accurate.

Picture the above design decisions being applied to the thousands and millions of events occurring in a large organization everyday and you get a sense of the enormity and complexity of TPSs.

The inner workings of a TPS are driven by the rules on which the organization runs. For instance, the inner workings of a hotel reservations system are dictated by the business rules and policies governing the reservation operation. Consider a hotel that has adopted the following policy: If a prospective guest cannot be accommodated in terms of the type of room requested, and if during the requested period a higher price room is available, then the guest will be accommodated in that higher price room while being charged the rate for the original room. The design of the reservations TPS of this hotel has to incorporate the above policy into it in order to work as desired. Not all rules are of an internal nature; they may be driven externally, such as by regulatory agencies.

There are few absolutes when it comes to the inner workings of a TPS. Business process reengineering has challenged age-old assumptions about business practices. Consider the practice of issuing invoices to customers. Some redesigned processes have made this practice obsolete. Or consider the practice of issuing purchase orders to suppliers. Continuous replenishment processes have made this practice obsolete too.

## IV. TPS AND DATABASE REQUIREMENTS

As pointed out earlier, a transaction captures the relevant attributes of an event for later use by other units in the organization or by external entities. This may be referred to as the *ripple effect* of transactions: Transaction data travel to all the other parts of the organization where they are needed. For instance, the news of customers returning the same defective product needs to travel to production, R&D, or purchasing, depending on where the cause of the defect may lie.

Transaction data are also used to update master files—where data about permanent entities (such as customers, products, suppliers, and employees) are stored. A customer returning a product will trigger a credit event that will update the field "credit available" in that customer's record in the customer file.

Because a transaction sets in motion the ripple effect discussed above, there must be a way to "de-ripple" those effects if the transaction is canceled. That is why a transaction is sometimes defined as *a set of operations that must be executed together, of which none are performed if any one of them is not performed.* In other words, a transaction either *commits,* meaning the changes are all made, or *aborts,* meaning any changes in progress are undone. For instance, in the example of a customer returning a defective product, if later it turns out that the product was really not defective (the customer just didn't know how to use it) and it is returned to her, then any updates that may have already taken place need to be undone.

A transaction is characterized by four properties referred to as the ACID properties: atomicity, consistency, isolation, and durability.

- *Atomicity* means that actions can be grouped together, and either all actions in that group will occur or none will. In other words, if some work is

started and a system failure occurs, the state of the system reverts to its state prior to the actions rather than the system being left in an intermediate, corrupted state. Consider, for example, the effects of doing only part of a financial transfer; a TPS should not be able to debit one account without crediting another.

- *Consistency* means that the transaction must represent a correct transformation from one state to another. For example, if a TPS credits one account in a financial transfer, it must debit the other by the same amount.
- *Isolation* means that actions do not interfere with each other. Multiple, simultaneous transactions must be made to appear as if they are actually a series of sequential transactions. If several users want to access the same bank account at the same time, the requests must be serialized and treated independently. Even if actions appear to happen simultaneously to the end users, they must be serialized and isolated internally.
- *Durability* means that once a change is made, it is not undone by a system failure. Changes to data must remain permanent once they are made.

Two of the most exciting uses of transaction data stored in a database are data warehousing and data mining. To avoid endangering an organization's operations, data are pulled from the operational system and placed in a separate data warehouse for the users to access. In data mining, mathematical modeling (in the form of statistical algorithms) is used to identify patterns in data and to analyze past transactions. For instance, data mining can be used to predict which customers are likely to switch to a competitor, or which transactions are most likely to be fraudulent.

## V. VIEWS OF TPS

Having explained some of the basic concepts associated with a TPS, we will now discuss ways of viewing a TPS.

A TPS can be viewed along two independent dimensions (Table I). First is the external/internal dimension. From the outside, a TPS looks like a machine that takes certain inputs and produces certain outputs. The external view captures events taking place between the organization and its external environment. To understand how those inputs are converted to outputs, one needs to open up the box and look at the internal view. The internal view captures events taking place within the organization

Second is the logical/physical dimension. The logical understanding of a TPS has to do with *what* information flows into/out of it, and *what* processes are performed on that information, and why. It is relatively stable and permanent. The physical understanding has to do with the technology used in entering the data, producing the information, and carrying out the processes. It is dynamic and transient due to rapid technological advances.

In what follows, we explore the logical and physical views and, within each, the external and internal views.

### A. External Logical View of a TPS

This view is concerned with the interactions between the organization and its external environment. We will focus on the interaction between an organization and its customers only. The interactions between an organization and its suppliers are just the mirror image of the interactions between the organization and its customers. The following is a general model of the information exchanges between an organization and its customers (Table II). It may be reminiscent of a game of ping-pong: For every action, there is a reaction.

Typically, the interaction starts with the customer's preliminary request for functionality (i.e., whether the organization produces the types of products or services that would solve the customer's problems), price, availability, and transaction alternatives (method of payment, delivery date, etc.). The orga-

**Table I** The Two Dimensions of a TPS

| | Logical | Physical |
|---|---|---|
| External | The information exchanged between the external entity (especially customer) and the organization | The technology (media) used for exchanging the information |
| Internal | The flow of the information within the organization | The technology (media) used for supporting the information flows within the organization |

**Table II** External Logical View of a Typical Sales TPS

| Customer → TPS | TPS → Customer |
|---|---|
| Preliminary request for information<br>Functionality<br>Price<br>Availability<br>Transaction alternatives | |
| | Preliminary information<br>Functionality<br>Product name/number<br>Price<br>Availability<br>Transaction alternatives |
| Order details<br>Product<br>Quantity<br>Transaction specifics | |
| Customer details<br>Name<br>Address<br>Financial details | Confirmation of receipt of order |
| | Confirmation of details |
| | Request for payment<br>Amount<br>Payment alternatives |
| Payment | Confirmation of receipt of payment |
| | Confirmation of shipment<br>Order number<br>Date to be shipped<br>Date shipped |
| Request for delivery status | |
| | Delivery status |

nization's response, of course, would be to provide this detailed information for the customer. Once the customer has done sufficient "shopping around," the order is placed. This involves the specification of order details:

- Product
- Quantity
- Transaction specifics (expected delivery time, method of payment, etc.)

as well as customer details (for new customers):

- Name
- Address
- Phone
- Etc.

The organization's response to this is typically a confirmation (echoing) of the details received. The order is then processed and, as a result, the following may be communicated to the customer: order number, expected shipment date, and actual shipment date.

Depending on the details of the terms of trade, once the requested shipment is made, there is a request for payment (invoice). In response, the customer makes the payment. In response, the organization may produce a receipt of payment. In the process of waiting for the shipment, the customer may want to inquire into its status. The organization's response will be to provide this information.

The above model is generic. As pointed out earlier, not every organizational TPS involves those transactions. In fact, the proponents of business process reengineering would argue that some of the elements listed above, such as invoice, are unnecessary and may be reengineered out of existence.

## B. Internal Logical View of a TPS

This view is concerned with the interactions within an organization that support the exchanges discussed above. It shows the various internal functions (processes) performed by the TPS as well as the data flows among them. As such, a data flow diagram is a suitable tool for showing the inner workings of a TPS. Figure 4 shows an example of the internal view of a TPS. It is meant to be representative, not complete. The major subsystems of this TPS are:

- Sales
- Shipping
- Billing
- Accounts Receivable (A/R)
- Production
- Inventory
- Purchasing
- Receiving
- Accounts Payable (A/P)
- Payroll.

As can be seen from Fig. 4, an organization's TPS typically consists of several interconnected cyclic systems with the output of one system often serving as input to another system. This interdependence serves to highlight the truly systemic nature of a TPS: What happens in one subsystem tends to impact other subsystems. For instance, if incorrect sales data are sent to billing and shipping, those functions are later haunted by returned goods and irate customers. The most integrated TPSs

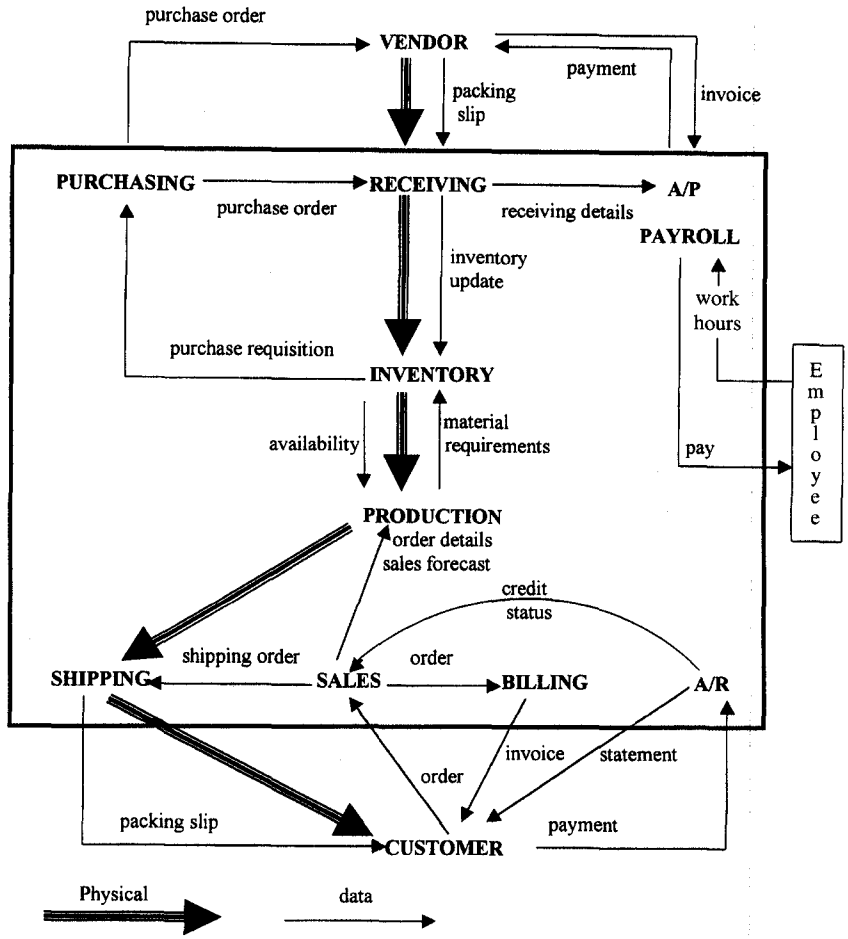


**Figure 4** A partial view of the internal, logical flows within a TPS. *Note:* In a retail environment, there is no production subsystem sitting between sales and inventory.

can be found in enterprise resource planning (ERP) systems, which aim at tightly linking the various subsystems of a TPS. ERP systems integrate not only the various operational areas of an organization but, more importantly, they integrate the operational level with planning and decision making needs.

Figure 4 is somewhat misleading in that it does not show the data stores corresponding to the various processes performed. The outputs from a process are not sent directly to other processes. Rather, they are stored in that part (file) of the database that is designed to store data indigenous to that process. Hence the sales process would update the sales file, the shipping process would update the shipping file, etc.

## C. External Physical View of a TPS

Recall that the physical view of a TPS focuses on the technology used in implementing it. Thus the external physical view is concerned with all technologies used in providing input and producing output. Rather than list the various input/output technologies, we will discuss the broader categories into which TPS input/output technologies commonly fall:

- *None.* This is the oldest TPS interface. The direct face-to-face communication between the organization and its customers without any technology used as the medium still works in some industries, such as retail.
- *Mail.* This is the second oldest model, where the traditional postal system is used for sending orders and payments, and receiving invoices.
- *Fax.* While very popular during the past 20 years, fax is a feasible TPS technology only in certain ways, such as sending in orders.
- *Phone.* This is also a traditional TPS technology that works more for the front end of the TPS cycle, namely, the customer's preliminary request

for functionality, price, availability, and transaction alternatives.

- *Electronic.* This is the most advanced form of communication. As such, it takes two different forms: free-format, and structured.
  - *Free-format.* E-mail communication falls in this category. Because e-mail contents are not governed by any standardized formatting rules, e-mail is not the optimal way of conducting customer-organization interactions.
  - *Structured.* Electronic data interchange (EDI) and Internet-based electronic commerce (whether business to business or business to consumer) fall in this category. In either case, predesigned procedures are used to regulate the electronic flow of information between the customer and the organization. When the customer-organization communication is completely structured, it can be automated. Automated customer service representative systems and automated teller machines are examples of this principle.

## D. Internal Physical View of a TPS

The internal physical view is concerned with all technologies used in converting the TPS inputs into outputs. In other words, they are processing technologies. Transaction volume is a major consideration in processor selection. An application that does not respond well to increased transaction volume is said to lack scalability. *Scalability* is defined as the ability of an application or platform to be expanded in terms of capacity to accommodate a larger number of users or transactions than originally planned without requiring significant changes in procedure.

Regardless of how technically advanced a processing technology is, if a TPS handles transactions one at a time immediately on request, then it is referred to as an on-line transaction processing (OLTP) system. The opposite of on-line transaction processing is batch processing. Instead of handling one transaction at a time, batch processing systems work on a group of transactions assembled for processing at a specific time, such as at the end of the day or the end of an accounting period. Payroll is a common batch processing application.

OLTP can also be viewed as a system of programs that facilitate and manage transaction-oriented applications, typically for data entry and retrieval in a number of industries, including banking, airlines, mail-order, supermarkets, and manufacturing. The classic example of a commercial OLTP product is IBM's CICS (Customer Information Control System).

Today's on-line transaction processing increasingly requires support for transactions that span a network and may include more than one company. For this reason, new OLTP software programs use client/server processing that allow transactions to run on different computer platforms in a network. Despite the recent shift to the client/server model, many legacy TPSs are still in use running on mainframe or midrange computers.

Regardless of the specific platform or architecture being used, the electronic medium has by itself revolutionized transaction processing. The difference between the electronic medium and the manual one is not just quantitative (faster, more accurate) but also qualitative (doing it versus not doing it). By hitting a single key on the specialized order entry keyboard of a point-of-sale system in a fast food chain, the order taker enters into the system a wealth of data (date, time, order details, employee ID, etc.) that are stored in the system for later use. Capturing this wealth of information in a low-tech, manual environment would not be feasible. A TPS that does not capture detailed data about the operation it is intended to support does not give its managers a clear picture of how effectively and efficiently that operation is performed.

# VI. STRATEGIC SIGNIFICANCE OF TPS

A customer's satisfaction with an organization is composed of two elements:

- The satisfaction with the *product* (service) acquired—its functionality/quality
- The satisfaction with the *process* involved in acquiring the product or service.

The latter is a function of the effectiveness and efficiency of the organization's TPS. An organization is typically represented in the form of a triangle/pyramid to show the management hierarchy. The TPS, being a subset of the operational system, is typically shown as the lowest level. Whether consciously or subconsciously, we associate more status and significance with higher levels and less with the lower ones. Most of those running the TPS, at the lowest level of the organization, are clerks of lower education and pay. But it is precisely with this level of the organization that customers interact in their day-to-day business dealings. It is this level—not the higher management levels—that customers "see" as the XYZ organization.

Even if the organization offers the highest quality product or service at the lowest price, the customers

have to go through the labyrinth of the customer cycle (inquiry, ordering, paying, shipping, etc.) before they receive the actual product or service. With the slightest dissatisfaction occurring in this preliminary interaction, customers may change their minds, cancel the order, and switch to a competitor. At its best, the strategic significance of the TPS lies in its ability to offer a pleasant purchasing experience to customers; one that would satisfy them enough to make them want to come back again and again. At its worst, the strategic significance of the TPS is revealed when it fails to work, thus bringing the inner wheels of the organization to a disastrous halt.

The strategic significance of a TPS goes beyond mere efficiency in that a TPS can incorporate features aimed at attracting and keeping customers. The field of strategic information systems is replete with examples of mundane types of TPSs that were given a strategic thrust. An example of a mundane transaction processing subsystem is order processing. A typical order processing system is based on the following logic. Order data are received, verified, and entered into the system. The system then verifies the customer's credit status and product availability. If both are positive, the relevant information is sent to inventory (or production as the case may be), shipping, and billing. If the requested item is not in stock, the customer is asked if he wants to backorder the item. If he does, that sets in motion a whole series of actions. Along the way, some exception reports and documents are generated also. Regardless of the complexity and number of steps involved, there is a clear-cut structure to this process.

If most order processing systems follow the above logic, then how can one stand out as strategically superior to others? The classic example of revolutionizing order entry was the American Hospital Supply Corporation's ASAP (Analytic Systems Automatic Purchasing). Order entry terminals were placed in hospitals and linked to AHSC's computers. Using these terminals, hospitals could place orders with a great deal of peace of mind, being assured of cost, availability, and delivery in advance. With a system offering this level of service, it is unlikely that a client hospital would switch to AHSC's competition.

With the increasing popularity of the Internet and electronic commerce, this model has now become pervasive and improved on by pioneers such as Amazon.com and Drugstore.com. The more value a TPS offers the customer, the more customers will be attracted and kept loyal. Therein lies the strategic promise of a TPS.

## SEE ALSO THE FOLLOWING ARTICLES

Accounting • Electronic Data Interchange • Manual Data Processing • Marketing • Operations Management • Procurement • Sales

## BIBLIOGRAPHY

Escalle, C. (February, 1999). Enterprise resource planning. *Harvard Business Review.*

Francalanci, C. (June, 1994). State Street Boston Corporation: Leading with information technology. *Harvard Business Review.*

Hammer, M. (July, 1990). Reengineering work: Don't automate, obliterate. *Harvard Business Review.*

Hopper, M. (May, 1990). Rattling SABRE—New ways to compete on information. *Harvard Business Review.*

Ives, B. (September, 1995). J.C. Penney: Fashioning a retailing nervous system for the future. *Harvard Business Review.*

Konsynski, B. (February, 1991). Baxter Healthcare Corporation: ASAP express. *Harvard Business Review.*

McKenney, J., and Clark, T. (March, 1995). Proctor & Gamble: Improving consumer value through process redesign. *Harvard Business Review.*

Nolan, R. (September, 1999). Drugstore.com. *Harvard Business Review.*

Rogers, T. J. (July, 1990). No excuses management. *Harvard Business Review.*