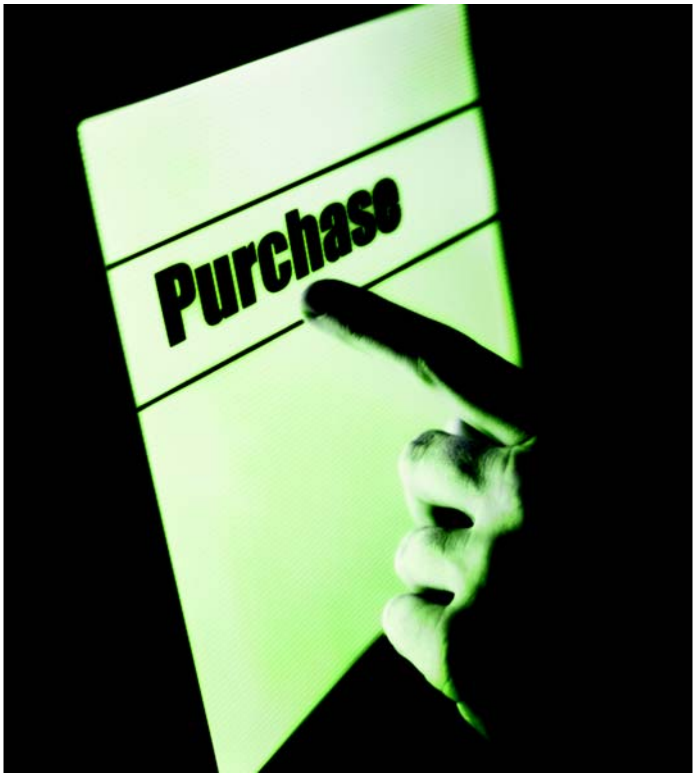
PART 2: THE TECHNOLOGY BEHIND SELF SERVICE

by Marvin Erdly and Amitava Chatterjee, CHTP



In a previous article from the fall 2003 issue we examined some of the drivers and value propositions fueling the adoption of self service in the hospitality industry. The second in a two part series, this article examines the technology that powers self service and makes it the least cost, and often preferred, channel of guest interaction. With hotel companies such as Hilton[1] and Starwood[2] working on pilot projects, it is clear that the industry recognizes the potential value of this important new channel.

This article reviews an application architecture model that is essential to drive adoption across the industry and to increase the overall value proposition. The key characteristics of this model include:

- An open platform architecture
- Support for multiple self-service channels (not just the kiosk)
- Support for multiple self-service applications (not just check in and check out)
- Adherence to the common use self service (CUSS) standard developed by IATA and ATA
- Robust systems management, including remote diagnostic support, across a geographically dispersed enterprise of kiosks

Solution Components

The software architecture to support this vision requires a three-tier model, as illustrated in Figure 1.

This article is Part 2 in a series discussing self-service in the hospitality industry. Part 1 **"The Case for Self Service in Hospitality"** appeared in the fall 2003 issue of *Hospitality Upgrade* on page 124.

Self-service Client Application

The self-service client application is the user interface appropriate for the channel. In the case of kiosks, this application would be written in an open language such as Java and would present the user with a simple, intuitive interface. Some characteristics of the interface include large buttons, limited text, self-explanatory navigation, multiple languages (if appropriate) and clear graphics to illustrate an action (such as a credit card swipe). The client application interfaces with device drivers that would control the underlying devices, such as a printer or card reader.

Self-service Transaction Server

The transaction server is a container for various process components that make the system work. It helps tie underlying applications, such as a hotel's PMS, to its self-service client application. The transaction server helps to implement the channel requirements and delivers the business logic and rules that the self-service application must follow. It performs two important functions.

First, it manages the interface to the underlying applications and converts transactions to the formats used by the channels of interaction.

Second, it implements the business logic which governs the workings of the self-service application (i.e., a guest cannot request a complimentary upgrade without having entered his or her loyalty program number). Note that this business logic should be limited to that required to support self service, as we recommend that the primary business logic remain in the PMS and not be duplicated.

Middleware

Middleware allows distributed applications to communicate with each other. By promoting standardization and interoperability, middleware provides an integration platform that allows the self-service solution to be extended

Kiosk Vendors Servicing the Hospitality Industry

Hotel Information Systems	www.hotelinfosys.com	Resells It Just Works Software for kiosk and wireless handheld units; solution can fully integrate with epitome PMS
IBM Corp.	www.ibm.com	Solution includes hardware and a custom user interface, which must be integrated with the PMS
Inter-American Data, LLC	www.iadusa.com	Hardware provided by IBM; LMS application can be extended to a self-service model
Kinetics, Inc.	www.kineticsusa.com	Solution includes hardware and a custom user interface, which must be integrated with the PMS
Multi-Systems Inc.	www.msisolutions.com	Hardware provided by SeePoint Technologies; PMS application can be extended to a self-service model
NCR Corp.	www.ncr.com	Solution includes hardware and a custom user interface, which must be integrated with the PMS

and scaled over time. For example, because middleware allows the kiosk application to be independent of the hardware, kiosk hardware and application upgrades are much simpler. We recommend that the middleware solution meet the IATA/ATA CUSS standard. This would allow the hotel self-service kiosk application to run on an airline kiosk, and vice versa (e.g., a guest can check out of the hotel and check in to the airline from the same device).

Systems Management

Network monitors provide automated monitoring capabilities that have the ability to manage and control a large network of kiosks by keeping an eye on the application and the hardware. They are able to link into enterprise systems management platforms as well. Key characteristics include:

- **Device monitoring in real time** – The nature of self service implies the absence of full-time face-to-face assistance. Because kiosks must maintain high levels of uptime, real-time device monitoring becomes crucial. Monitoring systems must have the ability to anticipate problems such as low key stock, configurable thresholds, a geographically dispersed enterprise of kiosks and notification of operations personnel via multiple channels.

- **Linkages with corporate system management platforms** – Such linkages allow a holistic monitoring of systems.

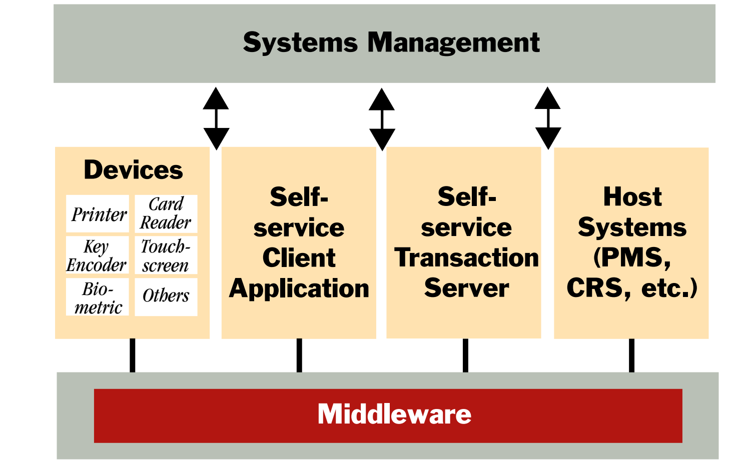
- **Usage analysis** – Monitors must track usage statistics and provide information that describes how the kiosks and the application are being used. Statistics on errors such as device outages are also stored for analysis and follow-up.

- **Scheduler support** – This allows for scheduled commands to be executed at the kiosk when managers determine, such as turning off the check-in application during peak departure periods, thereby freeing them up for check out.

Self-service applications are here to stay. Every day we read about another hotel company that has commenced a pilot program to implement such technology. We hear of airlines sharing their kiosks with others making it easy for modern travelers to fulfill their travel-related needs using a single device. Robust self-service applications are based on an open, three-tiered architecture framework, and include transaction servers, middleware and systems management. They help provide a feature-rich experience for guests.

Marvin Erdly (marvin.erdly@us.ibm.com) is an IBM Business Consulting Services Partner and Amitava Chatterjee, CHTP (amitava.chatterjee@us.ibm.com) is an advanced consultant in IBM's Travel and Transportation practice. Both are based in Fairfax, Va.

Figure 1. Three-tier Model for Systems Management



Notes

[1] *http://www.hotel-online.com/News/PR2003_3rd/Sep03_HiltonKiosk.html*

[2] *http://pacific.bizjournals.com/pacific/stories/2003/10/20/daily14.html*