



Integration between self-ordering and POS: A simplified explanation

by Juan C. Perez, president and chief technology officer, ADUSA, Inc.

- 17 Jul 2006

As self-ordering solutions proliferate in the restaurant/foodservice retail industry, there is one area that remains somewhat nebulous: how exactly does the integration between the self-ordering touch point and the POS work?

In order to understand this better, we first have to understand what a self-ordering touch point really is, so let's be perfectly clear – a self-ordering solution, whether deployed on an in-restaurant kiosk, a drive-thru kiosk, a tabletop kiosk, or any other touch point, is really just a customer-facing extension of the POS itself.

The retailer is likely to have a comprehensive POS system already in place (let's say from vendor A) when it is decided to explore self-ordering. Having reached this juncture, there are really two avenues that can be explored in order to find the right self-ordering solution:

1. Vendor A may already provide the self-ordering functionality as part of their POS offering. If the self-ordering functionality offered by vendor A is acceptable to the retailer, then this is probably the most sensible way to go since it is highly likely that the self-ordering function will already be integrated with the POS;
2. If the POS vendor does not provide the self-ordering functionality, or if there is anything else (function is not exactly what the retailer wants, it's too costly, retailer is already unhappy with support from POS vendor, etc.) that is not acceptable to the retailer with regard to the vendor's self-ordering solution, then self-ordering solutions from other vendors should be explored;

If the retailer exercises this second option and decides to acquire a self-ordering solution from a vendor other than the vendor that provides the POS, then the integration work will have to be done so that the two systems (POS and self-ordering) can properly communicate with each other. This is where the confusion can come in, because it may not be clear why these systems have to talk to each other.

Some software vendors talk about complex multiple layers of integration and warn retailers to be careful and ask the right questions (usually the ones that play to that particular vendor's strengths), but it is really not that complicated. Retailers should, at least at an operational level, expect to gain from their solution provider a complete understanding of what is involved. There are two major areas of integration to consider.

Integration from the self-ordering touch point(s) to the POS

Menu selections, ingredient selections, payment option selections, payments, in short, orders that take place on, for example, a kiosk, all have to be communicated back to the POS server via some programmatic interface usually referred to as an

API (Application Program Interface). The reason this particular piece of integration is important is that the self-ordering system is what is known as a "standalone" system and, naturally, the operational infrastructure (operations, inventory, sales, financials, etc.) is built around the POS. Therefore, every aspect of every business transaction that happens on the self-ordering system has to be communicated back to the POS from where the information can take its pre-established course and wind up as an order to be filled on the KDS, a balance on hand on an inventory report, a sales figure on a sales report, etc.

Integration from the POS to the self-ordering touch point(s)

Changes to the menu, price changes and suggestive selling scenarios are all examples of information that can be communicated from the POS to the self-ordering system. A good self-ordering solution will naturally have its own facilities for maintaining the menu, changing prices, etc., but what is the point of having to do things twice. If the POS already has a tool to maintain pricing of the individual items on the menu, then why not create an interface so that when a price is changed on the POS, the same change occurs on the self-ordering touch point? Again, a programmatic interface can be developed to pass the information from one system to the other.

That is really all there is to it. To be sure, there are technical questions to be answered, and these involve all sorts of potentially confusing technical jargon such as XML, asynchronous processes, data triggers, network traffic, etc. Good technical solution providers will be able to choose the right technical facilities and, more importantly, explain their rationale to the retailer in a clear and concise manner. The fact remains that integration between the two systems is a requirement, but it need not be made more complex than it actually is. As they say, "it isn't rocket science".