Eat, Shop, and Play: Mobile Computing Technology at Vindigo

Ramón Cáceres, Jake Donham, Bob Fitterman, David Joerg, Mark Smith, Tim Vetter
Vindigo, Inc., 259 West 30th Street, New York, NY 10001, USA
ramon@kiskeya.net

Developing widely useful mobile computing applications presents difficult challenges. On the one hand, mobile users demand intuitive user interfaces, fast response times, and deep relevant content. On the other hand, mobile devices have limited processing, storage, power, display, and communication resources. Vindigo has taken a technology-intensive approach to meeting these challenges and has created personal navigation tools that many people find useful in their everyday lives. Vindigo's Palm OS application, introduced in March 2000, has hundreds of thousands of users and continues to attract new users rapidly.

This article presents some of the technology behind Vindigo. To provide context, it first gives an overview of a typical user session. It then discusses special demands and constraints placed on mobile applications, and enumerates design principles drawn from these considerations. The rest of the article describes technologies developed at Vindigo according to those principles. Throughout, it emphasizes techniques that have proven effective in delivering location-based services on today's handheld devices.

User interface

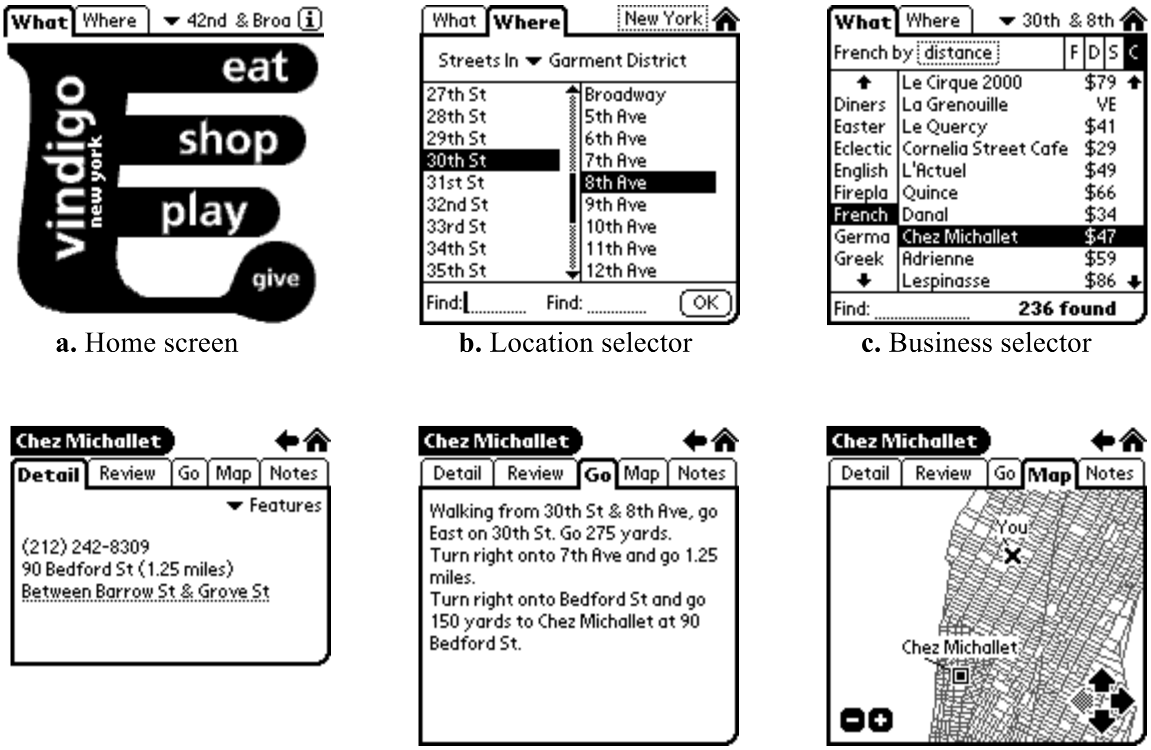
At the time of this writing, Vindigo offers guides to 20 major cities, including New York, San Francisco, and London. The guides provide information in three broad areas: dining, shopping, and entertainment. Content comes from established publishers such as The New York Times, The San Francisco Chronicle, and Zagat Survey. Vindigo's namesake guide application is available for Palm OS devices and Wireless Application Protocol (WAP) phones. A Pocket PC version is under development. Because of space considerations, the following discussion concentrates on the Palm OS version of Vindigo.

A user session begins with the home screen, shown in Figure 1.a. This screen provides access to the three main content areas (eat, shop, and play) and to a facility for wirelessly sending a copy of Vindigo to another device (give). It also allows switching between cities.

A typical session continues with the user specifying a geographic location. Figure 1.b shows an example of the location selector screen. The user chooses a street intersection by scrolling through two lists of street names. The lists can be narrowed by neighborhood. In addition, once a street in one list has been selected, the other list shows only intersecting streets. Defaults further speed up selection. For example, a neighborhood's central intersection is automatically selected when the neighborhood is first specified. Searching streets by name is also possible, in which case lists display only names matching a given search prefix. Finally, the application maintains lists of recently chosen and favorite locations for later use.

Vindigo also supports automatic location, for example by obtaining the current latitude and longitude from one of several Global Positioning System (GPS) receivers available for Palm OS devices. These coordinates are translated into the nearest street intersection, and the session continues as before. Automatic location is convenient in many situations, but there will always be a place for manual location. The main reason is that personal navigation is often about where a person will be in the future, such as when planning dinner close to home while still at work. Auto-location technology can only determine where the person is at that moment.

After selecting a location, the user normally proceeds to choose a category of business or event followed by a particular item under that category. Figure 1.c shows the category and business selector screen. Lists can be sorted by distance away from the previously selected location, by name, or by other appropriate criteria such as cost and rating for restaurants. Searching by name or name prefix is again possible.



a. Home screen **b.** Location selector **c.** Business selector

d. Business detail **e.** Walking directions **f.** Interactive map

Figure 1: Sample screens from the Vindigo Palm OS application.

Once an individual business or event has been selected, the application makes available five screens of information specific to that item. First, the Detail screen displays the telephone number, address, cross streets, and distance from the user. An example is shown in Figure 1.d. Second, the Review screen displays a textual review along with numeric or iconic ratings. Third, the Go screen provides walking and public transportation directions, which are calculated on demand and locally on the device. Figure 1.e shows an example of walking directions.

Fourth, the Map screen provides an interactive graphical map, as shown in Figure 1.f. The map provides simple pan and zoom controls, and uses gray scale to maintain clarity despite the limited resolution and lack of color on most current personal digital assistants (PDA). By default the map displays the locations of the user and of the selected point of interest, but very little else to avoid clutter on low-resolution screens. Street names are displayed when the user taps on a street segment, and names can be hidden by tapping on them. Maps can also highlight the route corresponding to the text directions shown in the Go screen. All map functions are also carried out locally on the device.

Finally, the Notes screen allows the user to record her own comments and ratings for later reference. These notes can also furnish important feedback to Vindigo and content providers, for instance that an establishment has closed.

In addition to the main user interface components described above, Vindigo offers a small number of menu items used to set user preferences. Example preferences are to which cities a user wishes to subscribe, and what distance limit should be placed on businesses and events listed in selector screens.

Demands and constraints

Mobile applications like Vindigo must contend with the demands of mobile users and the constraints of mobile devices. Mobile users have different needs from PC and Web users. For example, mobile users are more likely to be fulfilling an immediate need, not leisurely browsing for information. In addition, mobile users are more likely to be subjected to environmental distractions, not sitting in a quiet room. Because of these and related factors, mobile applications must hold themselves to a higher standard of usability than stationary applications. They must match the standards of successful consumer products: intuitive user interfaces, fast response times, and instant relevance.

At the same time, mobile devices have orders of magnitude less processing, storage, power, display, and communication capacity than stationary computers connected to the wired infrastructure. For example, the popular Palm III carries only a 16MHz processor, 2 MB of memory, two AAA batteries, and a 160x160-pixel black-and-white display. Similarly, wireless network services like Palm.Net and OmniSky deliver less than 20 Kbps of throughput while imposing several seconds of round-trip delay on every request.

Although wireless devices and networks are improving along all these dimensions, they will always lag behind their wired counterparts. It is also important to note that the processor speeds and memory sizes of mobile devices are improving faster than the bandwidth and reliability of wireless networks.

Design principles

To provide a high-quality user experience in the face of the above considerations, mobile applications must follow a number of principles. In short, they must:

- Offer intuitive and responsive user interfaces.
- Be customized to the form factors and capabilities of target devices and wireless networks.
- Exploit local processing and storage resources while making judicious use of wireless networks.
- Make intelligent use of context: location, time, schedules, contacts, and personal preferences.
- Provide deep, relevant content.

Research efforts to develop mobile guide applications have come to some of the same conclusions [1] [2].

Following the above principles, Vindigo has created commercial products at two extremes of the design space. One is an intermittently connected rich client for synced PDAs. This client maintains on the mobile device itself all the information needed to deliver the Vindigo service. It updates its information from a Vindigo server whenever the user synchronizes his device with a host PC connected to the Internet. Disconnected operation enables this client to provide full functionality and fast response times regardless of the availability or quality of network connectivity.

The other product is an always-connected thin client for WAP phones. In this case, the client consists of a standard WAP browser without any Vindigo-specific software. All information is retrieved on demand from a Vindigo server over a wireless link.

Vindigo is also creating a hybrid between these two extremes, a client for wirelessly connected PDAs. This client frequently connects to a Vindigo server to retrieve information over a wireless link. However, it also runs many user interface features local to the device to avoid the latency of going over a slow wireless network. Examples of local features are maps and directions. This client also caches and optionally prefetches data, further improving user-perceived latency.

Wireless connectivity gives the hybrid and WAP clients important functionality, like effecting real-time transactions, accessing much larger datasets than fit on a mobile device, and retrieving the latest information. At the same time, the hybrid client provides useful functionality and fast response times even when wireless connectivity is poor or nonexistent.

Vindigo technology

Behind these distributed applications is a large body of technology developed at Vindigo. Figure 2 shows the major functional components of client and server software. Because of space considerations, the remaining discussion focuses on PDA client functionality. Areas of innovation include:

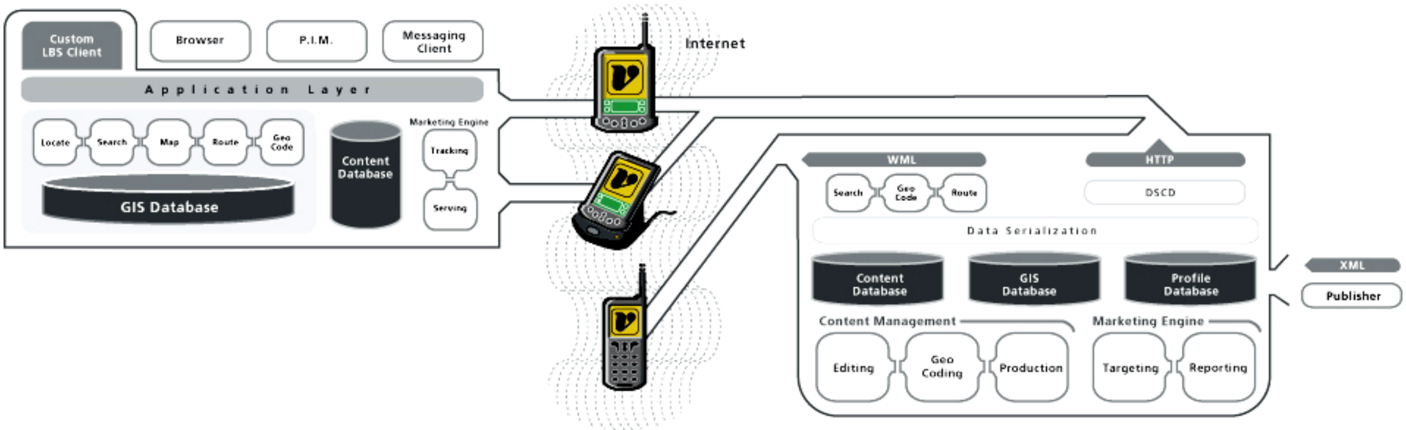


Figure 2: Major functional components of Vindigo software. In the center are represented the three types of Vindigo client: wireless PDA, synced PDA, and WAP phone. On the left are the components that run on PDA clients, and on the right the components that run on a server. WAP phones run a standard browser without any Vindigo-specific software.

Efficient text compression: Vindigo's text compression algorithm achieves compression ratios as high as 3.5 to 1. This algorithm is superior to standard ones for three reasons. One, it is designed to work in real time on mobile devices with slow processors. Both decompressing and searching through strings produces results quickly enough for interactive use. Two, it is largely context-free so that changing one letter in the text does not change all the compressed bits downstream. More specifically, local modifications, insertions, and deletions cause changes only within a 20-byte window. This feature is vital for efficient synchronization of differential updates, as discussed below. Three, it offers efficient random access. It can quickly retrieve an arbitrary piece of text without having to decompress the entire dataset, saving both processing and scratch-space overhead.

Compact representation of spatial and structured data: Vindigo compresses not only text strings, but also street maps, geographic coordinates, event schedules, business attributes, and telephone numbers. For example, the entire street network of Manhattan fits into only 52 KB, from which Vindigo locates users and businesses, draws maps, and computes walking directions.

Fast proximity searches: Vindigo's spatial and structured data, while highly compressed, allow for fast proximity searches on devices with slow processors. These searches, quite common in a location-based service like Vindigo, have been optimized for user-perceived performance. For example, a distance-ordered search over a database of several thousand items displays the first screen of results in less than one second while it continues to compute results to display on subsequent screens.

Differential synchronization of code and data: Vindigo optimizes the exchange of information between server and client to limit the performance impact of network access. It uses differential synchronization of code and data to perform byte-level patches, that is, only the minimum changes required to bring the client up to date are sent by the server. During a typical synchronization session, less than 5% of the total size of the synced PDA application is sent. An important benefit of simultaneously updating code and data is that there are never concerns about version mismatches between client and server. And as mentioned earlier, the text compression algorithm is designed so that changing a small part of the uncompressed text changes only a correspondingly small portion of the compressed text. This feature is critical for efficient byte-level patching.

Integrated server architecture: A unified server architecture provides information to synced PDAs, wireless PDAs, and WAP phones. The same architecture supports the user subscription website, customer service website, content management website, ad management website, geocoding engine, and directions engine. A common database access layer and location-based logic layer works across all wireless and mobile services. Similarly, a common wireless query server sends data to wireless browsers using XML over HTTP or WML over WAP.

On-device directions: Vindigo computes walking directions between the user and a point of interest on the fly, using the geographic data stored on the mobile device. It also finds the nearest public transit stops to users and to points of interest, and can generate walking directions to and from these transit stops. The routing algorithm was designed to produce results quickly enough for interactive use even on slow-processor, low-memory devices. Driving directions and directions within public transit networks are under development.

Responsive graphical maps: Vindigo also draws maps on the fly using data stored on the device. Map drawing is optimized to provide responsive panning and zooming despite the limited processing power of handheld devices. These maps are closely integrated with the location-based content in the rest of the application. For instance, a map is accessible after the user chooses a point of interest and it displays markers for the locations of the user and the point of interest (see Figure 1.f).

Wireless distribution: Vindigo can send a copy of itself via infrared to another device with a single tap on the home screen ("give" in Fig 1.a). A substantial fraction of users first obtain Vindigo in this fashion.

Simple user interface: Simplicity has driven user interface design throughout the application. As a result, Vindigo allows users to explore deep content with minimal interaction. For example, six taps with a fingertip are enough to determine what is playing when at the nearest movie theater, starting from the home screen and including selecting the user's location from a list of favorites. Even fewer taps are needed when reusing context established during previous sessions. In general, entering text with a stylus is rarely necessary.

Conclusions

The popularity of the Vindigo application attests to the effectiveness of Vindigo's technology-intensive approach to delivering location-based services on mobile devices. Hundreds of thousands of people have made Vindigo a part of their daily lives. Unsolicited feedback includes numerous anecdotes of how Vindigo has been useful in both personal and professional situations. As one quantitative data point, the average user consults Vindigo for directions once a week.

It is clear from the Vindigo example that mobile applications are fast becoming an integral part of everyday life. It is equally clear that, for these applications to be successful, they must be carefully tailored to the special demands and constraints of the mobile environment.

Acknowledgements

Many people have contributed to Vindigo technology development. Special mention is due to Aaron Ingram, Janet Kasdan, and Matthew Stecker for their work on the content and ad management systems, which fall outside the scope of this article.

References

[1] K. Cheverst, N. Davies, K. Mitchell, and A. Friday, "Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project," *Proc. of ACM MobiCom 2000*, August 2000.

[2] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyberguide: A Mobile Context-Aware Tour Guide," *ACM/Baltzer Wireless Networks*, Vol. 3, No. 5, 1997.