GettyGuide Resolution – April 2005

Executive Summary

The GettyGuide handheld architecture is extremely complex, involving numerous vendors and equating to multiple application and system interfaces. The complexity and demands of the architecture are such that it may never operate at a sustainable and acceptable level, without constant supervision from Museum IT and ITS staff. Some significant issues are confronting support staff, which is preventing the handheld from operating successfully, and despite continuous troubleshooting there appears to be no ‘magic bullet’ that will allow the handheld to re-launch in its current configuration. At the core of these issues is what is essentially an incompatibility between the location sensing demands and the delivery of streaming-audio content over the network.

Executive Recommendation

To achieve an acceptable level of ongoing operation, maintenance and support for the handheld, look to eliminate the wireless network for location sensing and audio streaming.

Wireless Network

The wireless network is being asked to essentially perform two functions, determine where the handheld is (location sensing) and deliver content to it (content delivery).

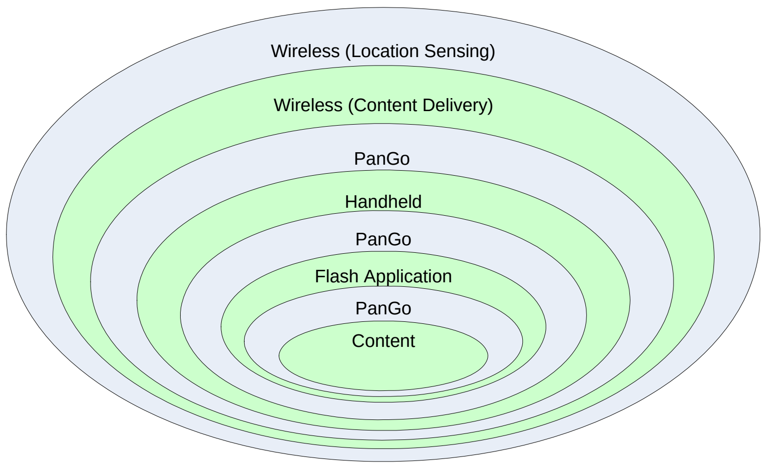
Location sensing works with the constant polling of the network by the handheld, to determine where it is. On top of this, the network has to deliver text, images and streaming audio to the handheld. There is an inherent conflict between the location-sensing piece of this architecture and the audio streaming, which is placing great demands on the network capacity and its ability to deliver. In simple terms, the mechanism by which the location-sensing calls are made demands a continuous priority on the network. Likewise, the streaming audio is demanding an equal and continuous priority, as it has to receive the audio in one uninterrupted stream. Multiply this conflict by the number of devices on the network.

The specification for the wireless protocol that the Handheld uses (802.11b), which is the generic protocol and nothing specific to the implementation at the Getty, functions optimally in a ‘burst’ mode. Think of this as a well-mediated discussion between many people – everyone is given their chance to contribute. The demands of the handheld architecture are closer to a group of people, none of whom will allow anyone else to be heard. As each handheld requests an audio stream, the problem is compounded.

Up to a point, the network can handle this demanding traffic. This is probably why the handheld trials appeared to be successful, particularly when independently testing content delivery and location sensing. In production, as the number of handhelds coming on to the network increases, the network becomes overly taxed to a point where one more handheld is effectively the last straw and everyone is affected. The manifestation of this may be either dropped or missing audio streams (the text and images may still be available), totally unavailable network connectivity for some devices, and/or lost location.

Furthermore, the effectiveness of the wireless network is impacted by other wireless personal devices on campus, such as some mobile phones, PDAs, and other wireless networks in the neighborhood. Unfortunately, the number and impact of these will only increase as time goes by as more and more people own them.

All these things, while they can be addressed up to a point, will demand a continuous and probably increasing requirement from support staff both in the Museum and ITS to maintain an acceptable level of operation. Furthermore, given the complexity and potential points of failure of the system, it is statistically unlikely to achieve a consistent operational level on par with that of the audio guide and one which would be acceptable to the Museum and ITS. By way of example, and this is an oversimplification, consider the integration points for the handheld system:



For example, PanGo has a software interface that communicates between the handheld's buttons and the Flash Application, as well as an interface that communicates between the Flash application and the location sensing network. These are integration points and therefore points of failure. Consider that if each one of these (eight) integrations has a 99% successful operational level, the overall operational level of the system is 0.99^8 or 92%. This is optimistic as well as oversimplified.

The only way to achieve a flawless operation is to remove as many points of failure as possible. By storing all content locally on the devices the statistical operational level essentially resolves to the reliability of the device itself, which can be substantiated at the point of exchange to the visitor. The only required support for a failed device is to swap it for a different one.

Storing the text, images and audio stops locally and switching off the location sensing, fundamentally changes the way the device operates. This change is a strategic decision for the Museum. One key piece of handheld functionality that requires network connectivity is bookmarking. This functionality requires minor network bandwidth since it is just the transfer of small chunks of text. With the major demands on the network removed, bookmarking would have a high success rate and is not core to the handheld's operation.

It likely that eliminating the network demands of location sensing and audio streaming, would allow text and images to be delivered over the network with an acceptable success rate in addition to the bookmarking functionality. Some extensive testing will corroborate this, which if successful would turn the decision about storing some content locally and some centrally, into one of ongoing maintenance code change requirements to switch to locally stored content.

To effect a change to a locally stored system would require three things:

LOCAL STORAGE – The purchase of additional hardware for the handheld to store the audio stops, text and images locally. The audio stops are currently available at a fairly high date rate (96kbps?). As it stands, this will not fit on the handheld but given the quality of the headphones and the nature of the content, there would be little perceivable degradation in quality if the audio were sampled down to a level that could fit on some currently available storage. It would be a simple test to judge the acceptability of this re-sampling. There are a number of solutions for additional storage for the handheld starting at approximately $65 per unit for a 1Gbyte Compact Flash card. This may be sufficient. In the worse case, greater storage could be purchased, e.g. a 2GB card at $160 per unit. For 500 handhelds, the best case would be approximately **$30,000**, the worse case **$80,000**. Volume and/or educational discount may reduce these figures. Note that the Museum would only have to purchase as many units as it needed for the public at any one time. These storage cards have no moving parts, great reliability and can be swapped easily between devices. If money is an issue the Museum could consider purchasing a smaller number, say 200 (**$13,000**), and only make this number available to the public.

FILE SYNCHRONIZATION - Implement a mechanism to file-synch the devices when they are docked. Essentially, a master version of the application and content is maintained centrally and the handhelds are scheduled to synchronize periodically based on the frequency of content change. There are a number of mechanisms that could be employed, a ‘push’ (similar to what ITS does to upgrade our desktops) or a ‘pull’ whereby the Handheld initiates synchronization when it is docked. On a daily basis this synchronization would be purely event-related information, which may only require the transfer of a single text file. On a weekly or monthly basis this would be revised or new collections-related content. A small amount of research reveals third-party products retailing at around $20 that will do this for a single configuration, but any product would have to be configured for the Museum’s multiple-configuration architecture - some research would need to be done. An ideal resolution would be a site license for one of these products, installed on each handheld, pricing to be determined. In the worse case a smart programmer for three weeks could probably build a file-syncing application. Best case, assume 15 days at $680/day = **$10,200**, worse case 25 days or **$17,000**. Alternatively a flat fee could be negotiated. Some research will need to be done into the logistics of setting this up, in the worse case some additional hardware may need to be purchased, hard to quantify without further research. Alternatively, the synchronization could actually be done over the wireless network out of business hours.

FLASH APPLICATION - Recode the Flash application to work locally. This would require functional changes to allow the user to specify their location, and application changes to eliminate the wireless access and grab text, images and audio from the local card. This is hard to budget for without some review of the way the handheld data is organized and the way the application is built. Some time would need to be spent prototyping how the changes could be implemented with regard to the user interface. Ball-park figures:

- Prototype: 15 days at $680/day = $10,200
- Build: 25 days at $680/day = $17,000
- QA & Testing: 10 days at $680/day = $6,800

$34,000

Summary – These figures are based on a cursory review of the architecture, and some initial thoughts on how to proceed. The hardware elements are easy to quantify, the development costs less so. Totaling these figures gives:

Local Storage	**$13,000**	**$80,000**
File synchronization	**$10,200**	**$17,000**
Flash application	**$34,000**	**$34,000**
	$57,200	**$131,000**

These figures are capitol costs and do not address the hidden costs of time and resources borne by internal staff.