

***m3i* – Concepts, Goals & Aspirations**

***m3i* – Man Machine Mobile Interconnect**

All of us are different. Some of these differences, physical disabilities for example, can profoundly affect how we interact with machines but often, even when the differences are more subtle, the effect can still be surprising. Many of us know people who have found it difficult to program a video recorder or who do not find it easy to work a PC. This is not a function of intelligence. They merely do not understand the User Interface (UI) metaphor that is employed. This understanding comes from having experienced similar things before and being able to predict the action expected by the designer of the machine and its UI.

There are a variety of reasons why some users may be less fluent with a machine's controls than others. These may be physiological, may derive from experience or expectations and some may even arise because of environmental issues or peer pressure. Additionally, the same type of machine but from a different range will employ a different UI, further complicating matters. Worse still, the UI is usually designed by the people who designed the machine itself and who, by definition, think about the machine in terms of its detailed functionality. Consequently, they tend to manifest that detail in a UI. After all, why would they hide it since they had created it? It is also not helpful that products tend to be marketed on feature sets, rather than ease-of-use.

Through our experience with accessibility solutions and ease-of-use, we began to realise that interacting with machines via a shared user interface physically within the device is probably at an evolutionary dead end. Addressing this is about managing user confidence and that is boosted through the UI experience.

Goals

UIs are not about machines, they are about users. Our long term goal has been to position the UI where it belongs - with users. We want to give them a consistent way to interact with the same type of machine and to provide UIs that work for users, not UIs that work for machines.

m3i is about connecting people with machine functionality on their terms, from wherever they are, through various UI device metaphors: how a UI manifests; how it works; what functionality it reveals; its physicality; to be able to use multiple UIs in concert; to meld functionality from different machines into a seamless process; to be able to abstract functionality to a higher level; to be able to (seemingly) expand or recombine the functionality of a machine through external services. All these elements empower users to have control over their UI which, ironically, mentally connects them to the functionality of the machine on their terms. This in turn leads to confidence and ultimately more use of the machine which, after all, is the intent of the original machine designers.

m3i is a platform. Its purpose is to encourage developers and integrators to build innovative ways of working with machines to the benefit of their target users. Select Technology will, of course, also develop its own products using the platform.

Steps

Select Technology has been making User Interfaces (UI) for photocopiers and, more recently, Multifunctional Devices (MFDs) for many years. Recently a variety of enabling technologies became mature enough to allow us to move forward.

MFDs were the first machines we chose to *m3i enable*. This was because they are functionally complex, there are known issues with users going beyond basic functionality and we have direct experience having worked with them previously. We chose Ricoh's range of MFDs due to the rich embedded environment available.

One

The first step was to expose the machine's functionality in an open-standards way across a network by developing and installing a Machine Interface (MI). There were certain presumptions that we made at the outset. Networks will be ubiquitous, all devices will eventually be networked and Open Standards encourage developers to use your platform. The technologies employed by the *m3i* platform are detailed elsewhere. Briefly though, we employ HTTP/S as a transport for Control and asynchronous Monitoring of the MI from clients. This is the base *m3i* transport and *miRicoH* itself employs a variety of standard technologies to enable it to provide its services.

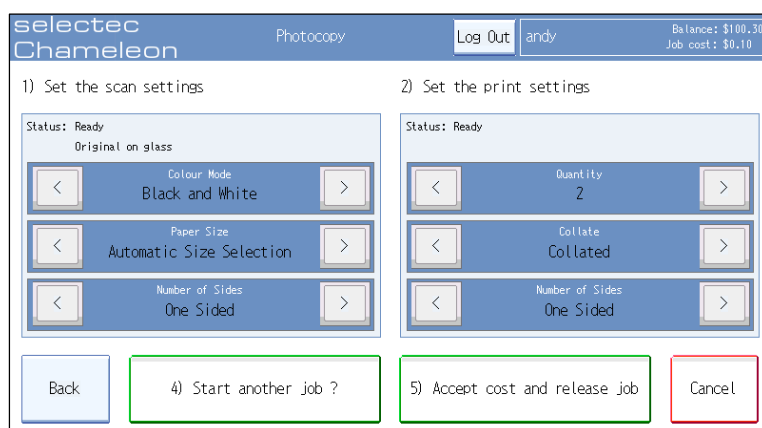
Two

The next step was for us to build some UIs to consume this functionality. The idea was to create some UIs that not only addressed specific problems and would therefore become saleable products, but that would also serve to inspire developers and implementers to employ our technology to solve their problems. To this end we conceived of three UI types and built a variant of each :-

MyUI is a dynamic browser based UI that is hosted by *miRicoH* in the MFD. The first variant of this is the Access Module which is designed to address ease-of-use and accessibility. This was specifically designed to support accessibility aids e.g. screen readers, and to run from small screens. It can be used from PDAs, PCs, Cell Phones, in fact, anything with a compatible browser. The idea was to build a UI with a simple usage model such that it could be used as easily by a wheelchair-user or visually impaired person as by anyone else. We purposely did not set out to create a rich UI experience as this would have complicated accessibility compatibility. We employed basic technologies like DHTML and JavaScript to deliver a consistent experience on different devices and to support as many accessibility aids as possible.



Chameleon is the logical evolution of our earlier Chameleon and Ease-of-Use Module products. These were external hardware devices which allowed a customised screen and functionality to be defined for a particular installation. We are now able to host this inside the MFD's LCD/touch-screen. Chameleon screen sets are created using an XML file that describes the layout and the actions to take when buttons are activated or events occur. We have example screens for Chameleon that anyone from end users to developers can adapt



for their requirements. One should note that even though the Chameleon is hosted within the screen of the MFD that it is controlling, it does actually interact with the MFD via *m3i* over the network. For this reason its data is the same data as that shown on MyUI and, of course, the two can be used together if required. Also Chameleon is able to use additional *m3i* services, if present, that are not in the MFD itself.

MyVoice is a server based voice application (as a hosted service or on the customer's network). A VoIP handset is commonly attached to the MFD such that when users pick it up they are unaware of the server interaction and feel like they are having a 'conversation' with the MFD. As users are speaking into a handset in a natural, bidirectional fashion, they feel comfortable speaking to the MFD and are happy to interact in this way. Of course any other visual UI, like Chameleon, can be employed to further enhance the experience by providing detailed visual feedback to complement the conversation or allow non-voice instructions to be issued if preferred. Incidentally, the voice conversation with the MFD can be delivered on any SIP compatible device, which includes software based ones on cell phones. We envisage a situation where users will walk up to the MFD, gesture their cell phone at it, have a personalised visual UI appear on their screen and be able to speak to the MFD naturally through their phone.

While we have grounded these examples in the MFD area one should remember that this equally applies to any machine that becomes *m3i* enabled.

Three

Stage three encompasses extensibility. We have a platform that describes abstract machines in a consistent fashion across a network such that a variety of UIs can interact with those machines. That is fine for a systems integrator who needs to customise a machine for a particular customer's requirements, but not enough to interest developers to integrate external functionality. To aid this, we have allowed for any Machine Interface to be extended to include additional attributes beyond those offered by the original machine. We also allow for jobs processed on the machine to be extended with additional data. We have even provided generic jobs, so that the MI will seem to process a job that is actually managed by an external entity but appears part of the normal job set as far as UIs are concerned.

To show this in action we developed two further components. While a developer is free to interact with *m3i* at a low level, we felt that there was a need for a generic connector that removed the complications of connecting to and monitoring Machine Interfaces, so we created *m3iConnector* for .NET. Then we proceeded to build an example of how to integrate a Copy Accounting system into an *m3i* network using this connector. This is the most complex example we could think of and it demonstrates many of the required techniques. We expect many other integrating issues will be trivial in comparison. If, for example, one wanted to collect some items of metadata when scanning to a Document Management System (from any of the UIs, including MyVoice) this would be relatively easy.

Aspirations

m3i with its Machine and User Interfaces is a very easy way to deliver functionality from machine to user in a variety of ways. Add in the customisability of the User Interfaces, the extensibility of the Machine Interfaces and the incorporation of network services and it is an extremely flexible platform for systems integrators and application specific developments.

Now that we have a working system, the goal is to work with partners to develop innovative applications and UI types. We have aspirations in a number of areas. For example, we plan to target the iPhone as a UI and probably to create a richer browser UI using Silverlight. There are a number of back-end hosted services that will be delivered to complement the existing system. Development is also required to incorporate new seamless ways of identifying users so that Interfaces and services can be tailored to the individual user. All of these things are immediately available to *m3i* customers as soon as they are developed as the underlying *m3i* platform is, of course, consistent.