Towards an Information-Centric Electronic Society

Ing-Marie Jonsson
CSLI, Stanford University.
Stanford, CA,USA
ingmarie@csli.stanford.edu

Qamir Hussain
School of Computer Applications, Dublin City University.
Dublin, Ireland
qhussain@compapp.dcu.ie

ABSTRACT

User Interface conventions generally force users to learn how the application works rather than allowing users to interact intuitively. Often, the application is designed from the perspective of the underlying technology rather than highlighting the offered functionality in an intuitive manner. Well known examples include, the VCR and the answering machine where power users are restricted to engineers and users that likes to read manuals. This issue isn't restricted just to the household devices**;** similar shortcomings can be found in other electronic devices. The usefulness of the best software can be hampered by inadequate or badly designed interfaces.

Questions that are discussed in this paper are: The existence or creation of general principles to support intuitive interaction with devices and applications. The use of possibly outdated metaphors for the creation of user interfaces that potentially restricts presentation or information, and interaction with information.

Presented in this paper is a design methodology that addresses these issues by focusing on the properties of information rather than the properties of user interfaces. This approach has the potential to enables designs that can cater for every possible user and different type devices.

Keywords
Human Information Interaction, agent oriented approach, design methodology, interaction models.

INTRODUCTION

Users need to be presented with the option of viewing information in a variety of formats coupled with flexible interaction methods to get at the information more effectively. This can only be achieved through the use of mantras that are well known and used in design schools, but seldom highlighted in computer science; in essence these mantras try to instill the idea that form follows function. Restricting User Interfaces (UI) and interaction style to one view of the information is counter-intuitive. This ultimately restricts the user to a way of thinking that may not be the most suitable. Multiple views must be supported to provide sufficient flexibility in matching individual interaction styles. The design of these interaction styles entails the creation of UIs that crosses cultural boundaries, accessibility, cognition, perception and human information interaction [1].

Fundamentally the user needs the ability to view the information in a variety of formats and the ability to select suitable interaction methods. The first step to achieve this goal is to move away from the paper and document centric view of electronic information, ordered as files and folders in your device. Information needs to be the basic unit of currency in electronic environments. This frees designers of systems to focus on how to interact with information at a fundamental level and thus allowing for both a more flexible information exchange between applications and for richer conversations with the user.

This should allow for a more finely grained modular system with inherent properties that supports multiple formats for Information, multiple views and multiple interaction styles. Included in this approach is that electronic devices carry knowledge about their properties, capabilities and limitations. This is essential since Information might not be feasible to present, or even more importantly, not legible when presented on certain devices. As an example, try presenting a video stream on a device without screen.

The approach outlined in this document is based on Information as the basic unit and it is implemented using an agent-oriented system. Individual agents in the system perform restricted tasks, for example presenting information in a particular format and supporting a particular style to interact with that information. These agents (the information set and interaction agents) communicate with the device to provide the interaction style and presenta-

tion style best suited for the device and the particular user.

THE RELATIVE PROGRESS OF COVERGENCE

Transparent information access still has many barriers to go through. The right tools need to exist before this transparency is realized. These tools need ultimately to be flexible and simple to use. The basic foundation for most applications lies in the file or document model of information storage. These models inherently create barriers against smooth information retrieval especially with large information and/or when much cross-referencing is required [3].

The convergence of devices has been talked about at least since the inception of the world wide web, PDA's (Personal Digital Assistant) and mobile phones have become commonplace with lines of functionality blurring. Standards and capabilities of smaller devices are extending and becoming more open. It is almost possible to visualize the all-in-one wonder device, combining the flexibility and functionality of a desktop computer with the mobility and ease of use of a mobile phone. Bandwidth is becoming less of an issue for this all-in-one convergent device. Displays, speakers, microphones and haptic technology are not too far behind in terms of the ability of these technologies to provide the user with sufficient access to information.

In reality, the progress of convergence is hampered in part by problems in information currency, many of which revolve around proprietary issues. It may seem obvious to say that information is the basic currency of the electronic world; however this is not reflected in the design of User Interfaces. The application and document-centric approaches both fall short of providing direct access to information [3]. These approaches inherently place barriers around information contained within the file or document structure. The result being that access to information within these models ends up as a closed process.

WHAT YOU WANT AND WHAT YOU GET

Greater flexibility and increased functionality can be built into the User Interface at an earlier stage by making information the basic currency and thereby operating an information-centric approach [2][4]. This allows for the ability to move information between applications more transparently, greater flexibility in cross-referencing information and easier retrieval of information.

Currently, transporting information around the Windows Desktop relies heavily on clipboard mechanisms due to the document-centric nature of the platform. A document-centric or application-centric approach provides little to no flexibility to allow alternate views of information and alternate interaction models. This becomes especially difficult to achieve when confronted with large information loads.

The document-centric approach also places restrictions on transparency of information; relationships in related data may not be seen due to an inability to easily view information from one application in another.

Applications that keep information encapsulated within objects and documents create inflexible and awkward interactions with the information.

As an example: use the Drag and Drop metaphor advertised as part of Microsoft® Windows 2000™ and try to drag text from MS Word™ and drop that text into Notepad™. At this point you realize that the scope of the Drag and Drop metaphor does not cover this particular functionality. So, what seems very intuitive becomes frustrating until you learn “how it should be done”, that is, what mechanisms these applications support for you to manipulate documents and text within these documents. For these applications, the manufacturer has selected Cut and Paste as the appropriate mechanisms. The use of the clipboard for inter-application communication, in this case to move text between two text editors, is both cumbersome and non-intuitive. Especially when moving text within the same application is supported by the simple Drag and Drop metaphor.

The adoption of an information-centric approach means that the tailored import and export mechanisms as well as the clipboard mechanisms are bypassed in favour of a more flexible approach [4].

A consistent and thereby more intuitive use of the Drag and Drop metaphor can be seen in the Visage software environment [2]. Visage supports different visualization strategies for pieces of information and objects by simply dragging and dropping these pieces of information and objects between applications in the environment.

FOCUS ON INFORMATION OBJECTS

The value of a device is in its ability to give access to information through its applications. Information is a commodity and should be treated as such. The authors advocate as shift towards design with the focus on interacting with information and not the device; the device is merely a conduit for the information.

This approach, the information centric approach, signals a shift over to the design of information and suitable visualizations instead of widgets and buttons. For this to happen, the approach based on the concept of Human Computer Interaction (HCI) must be replaced with the concept of Human Information Interaction (HII), an approach where information, not the computer, is the target of interactions.

HII is defined as "how human beings interact with, relate to, and process information regardless of ... the medium connecting the two" - Nahum Gershon [7]. After all

isn't that we expect to happen and what we are doing when we press enter?

INFORMATION CENTRIC ENVIRONMENTS (ICE)

Visage is an example of an information centric approach, and it implements a UI environment for exploring and analyzing information. With this approach, Visage tries to remove barriers of direct user access to information across applications and different visual representations. The goal achieved in Visage is that information moves from one visualization to another transparently (for a more in depth analysis see [2]).

The information centric approach advocates that a single application should not restrict the view of an information type. Furthermore, the approach supports mechanisms to allow information to migrate transparently from application to application transparently. This enables easy access to different views and visualization of information. This is a handy tool to use when designing applications that are to be accessed from different type devices.

Design Principles for ICE

Design principles will be needed to realize frameworks and applications within the information centric paradigm. These guidelines are based on the definition of information as the basic unit of the UI, and the principle of transparency of information. The guidelines addressing mechanisms needed for information intensive domains are adapted from Visage.

- Information is the basic currency and the focus of design should be on suitable presentations of information types.
- Information needs to be decoupled from the conventional viewer [9]. It should be possible to build viewers that adapt and limit the presentation of information types to match restricted devices.
- Information and interactions needs to be decoupled from applications [9]. It should be possible to adapt the interactions with information to the device it is accessed through. The results from asking for directions while sitting in front of a desktop machine might be a list of driving instructions. The result from the same application when accessed via a telephone might be a dialogue with a live operator that relays the same information, or a speech-based interface that allows interactions with the requested directions.
- Techniques needed to select and combine information from multiple application interfaces.
- The ability to enable rapid generation of visualizations to integrate information.
- The ability to control filtering and level of detail and create new information wherever its displayed.
- An environment should make it easy for people to share and communicate their results in collaborative settings, where they must iterate between analysis and presentation activities frequently.
- Work with the trade-offs in design between single application and multiple application approach to support access by multiple devices.

These guidelines make it evident that there is a mismatch between the ICE approach and the conventional file system and storage approach. If we dispense with the hierarchical desktop idea, are there alternative that better match the ICE approach on how to store information objects?

ORGANIZING INFORMATION OBJECTS

The best match for storage of information objects is a strategy called lifestreams[6,7]. A lifestream is a chronologically ordered stream of discrete information objects, (known as chunks) which can be treated as logical units. This logical unit can be a piece of information such as an e-mail, a calendar entry or an audio-clip, and a lifestream has the following properties:

1. File names and folders are abandoned as organizing devices for information chunks.
2. There exists a sophisticated logic for finding a chunk or groups of chunks.
3. There exists logic for summarizing or compressing a large group of related chunks.
4. Compatibility (across platforms) is automatic.
5. Explicit file storage is to be abandoned.

Points 2, 3 and 4 should be self-explanatory, however, 1 and 5 need further investigation at least in the context of this discussion.

The creators of Lifestreams found the concept of file and folder to adhere too rigidly to the paper filing system of the desktop metaphor. A file sits in one folder, that folder in another, and so on. Relationships between files in folder A and files in folder B may never be realized because of the limiting nature of the filing system. The name of a document doesn't necessarily hold much meaning, especially for retrieval purposes, it is within the content that the value of a file lies. By abandoning explicit file storage, the door opens for mechanisms that provide anywhere-anytime access to one's own information [6,7].

Lifestreams represent a different and powerful approach to the storage of information objects. Its inherent non-linear and non-hierarchical nature enables rich individual interaction styles, at least conceptually.

The approach offers an alternative to the desktop's method of hierarchical storage. Even though the idea is conceptually extremely interesting, it is a major effort to adopt the strategy due to the lack of support by basic tools to access the information objects

In our approach to ICE, information storage is decoupled from the applications. This enables portable solutions to be implemented, however, given the opportunity to choose an information storage approach, lifestreams presents a well-matched and appealing choice.

AGENT ORIENTED APPROACH TO ICE

A prototype system based on the design principles listed for ICE was implemented using a software agent technology. Solutions implemented in the selected agent technology are configured as arbitrary networks of collaborating agents. Agents collaborate by messages using a strategy based on delegation and claims [10].

Information and Interaction Agents

Each individual agent in the system is responsible for a restricted task. As an example, one agent (or rather network of agents) is responsible for a particular information format, this agent will provide mechanisms to present that information format as well as functionality to interact with that information. Each such information and interaction agent, is also responsible for collaboration and communication with other types of agents to provide the interaction style and presentation style best suited for the device and the particular user.

Device Agents and Application Agents

To support the concept of ICE, applications are decoupled from input and output devices. The system instead relies on agents that manage profiles to provide personalized and adapted interaction and presentation styles. Currently in use are three types profiles, device profiles, application profiles and user profiles.

Device profiles contain information about capabilities and limitations of the device. Application profiles also contain information about capabilities and limitations of the application, and finally the user profile contains the user's preferences and requirements.

Input /Output Modality Agents and Filter Agents

Input and Output Modality agents are located on devices and servers on the network. Each of these agents is responsible for implementing a particular modality and also for communicating the stream to other agents in the system. There are two types of modality agents in the system, simple modality agents and complex modality agents.

Simple modality agents are directly connected to a I/O device whereas complex modality agents will receive input from multiple simple modality agents. The complex modality agents will combine and interpret these input streams to produce the I/O modality.

There is also a class of agents, often located on the network, that handle manipulation of information such as transformations, sorting, and filtering.

Accessing and Interacting with Information

The decoupling strategy that separates applications from storage strategies, applications from interaction models, and finally applications from information formats as much as possible, allows for a dynamic system.

This enables us to manage a dynamic configuration of presentation formats and interaction methods, based on the capabilities and limitations of the application and the access device. So when a user accesses an application, the underlying system initiates the negotiation needed to configure the interaction. The negotiation is based on the information contained in the device profile, the user profile and the application profile.

MOVIE FINDER APPLICATION

Movie finder is a sample applications implemented in ICE. The application provides information in response to inquiries about movies, and three device/interface/information combinations are depicted and described.

The application supports natural interaction and typical inquiries to the system are:

- *Give me a list of action movies playing in San Jose tonight?*
- *What is the address of the AMC theater in Santa Clara?*
- *Are there any movies starring Tom Hanks playing Saturday evening?*

The resulting information can be presented as text, graphics, video and audio. The matched device and information interfaces are all based on combinations of the following information:

Movie Theaters: The name, location, current movies, and telephone number for each movie theater.

Movie: The name, starring actors, genre, short description, video clips, pictures and audio clips.

Movie Stars: Name, Movies, Acting History and Trivia

Movie Information on a Laptop

The first device was a laptop with a 14" screen where the resolution was set to 1024x768. The laptop had a keyboard and a two-button touch-pad. The device was equipped with a 10Mb Ethernet connection to the application server.

The movie information system designed for desktop makes use of the entire screen real estate and displays enough information to give the user the perception of

viewing a large amount of information without being overwhelmed.

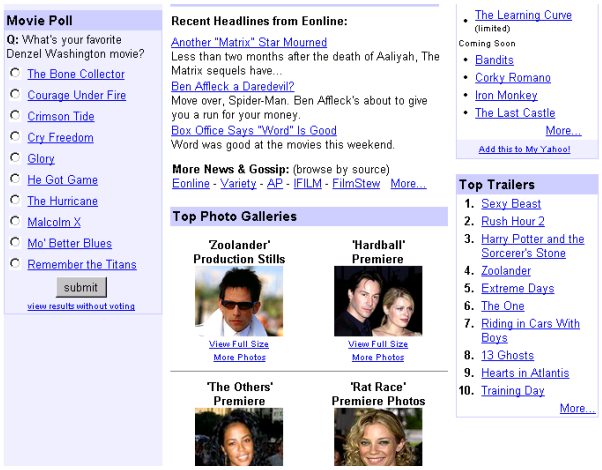


Figure 2: Movie Finder on a Laptop

Movie Information on a Hand Held

The second device was an iPAQ with a 3.7" screen with a resolution of 240x320. The iPAQ had a touch-screen pen interface; and an on-screen keyboard operated with a pen. The device communicated through a 11Mb wireless network connection to the application server.

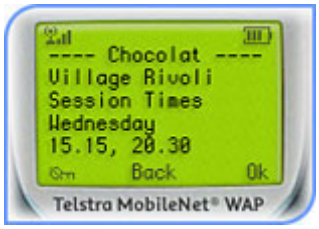
Depicted below is the same Movie Finder application presented on an iPaq handheld PC. The screen real estate is used to display enough information without having the screen cluttered and causing confusion.

Figure 3: Movie Finder on an iPAQ

Movie Information on Phone

The third device was a Nokia 7110 with a high-contrast 96x65 pixel display. The 7110 had a keypad and a navi roller equipped with predictive text input. The phone communicated with the application server using a 14.4 kbps connection to the application server.

Finally we see the application as presented on a mobile phone. Due to the extremely restricted viewing parameters the information is presented in a text only format. Although it may seem that very limited amounts of information is displayable; it has been filtered by the system through series of successive interactions by the user to reach the information for the movie now presented.



Discussion

This experiment with the same application presented on three devices highlights the importance of focusing on interaction with information. It is clear that there are obvious data types that are not suitable to present on certain types of devices. This is for example, a video clip presented on screen-less device and an audio only interface on a large-screen device. The first example is an instance where the data cannot possibly be understood; the second example highlights an instance where the properties of the device are under utilized.

There are also other less obvious mismatches: it is hard to present tables, graphics, and video on an iPAQ, and it is almost impossible to present these data type on the Nokia phone. An obvious follow-up is to study information approximation by using alternative representations for these data types and to look for indications when the burden on the user is to high, and the message is not worth the price of struggling with the decoding.

There are of course many factors that impact the thresholds, such as, the relative importance of the information, the available alternatives, and the user's experience to mention a few.

Another important result from the experiment was the impact of controlling the volume of information presented to the user. Here there are also obvious mismatches, one where the user was presented a large list of movies with all associated information over the phone, and a situation where the user was presented a three-word result on a desktop system.

Varying the volume of information is a less common variable in interface design. It has mostly been an area of interest for designers that focus on applications accessed over the phone. A follow-up here would be to study how the difference in interaction style affects desktop users. If an interaction style using more interactions and filtering at the application end is accessed via the desktop, will the change in style and the limited use of screen real estate be perceived as negative?

SUMMARY

The ultimate aim of our ICE approach to pervasive computing is to provide interaction with information intuitively and without boundaries imposed by applications and/or devices. Flexible interaction styles are those that allow users to interact with information effectively whether through a workstation, PDA or VCR.

Transparent computing will arrive sooner if information-centric approaches are adopted. Information objects need to be designed to reflect how people interact with information. Applications and devices need to be flexible in order to allow individual interaction styles to take place. Flexibility in a system reflects the nature of the information that the systems handle.

If information objects are not designed well enough, interaction styles will be less meaningful and the full value of the information will not come through. Not giving users the ability to see the full picture only shortchanges them.

The increase in the mobile work force, where alternate devices are used to access the same application, constitutes a strong driving force to shift over the information centric approach. Access to information is important and whatever device is used for access should support the user to gain the requested information instead of hampering access by trying to display or present information and devices that lack the ability to do so

Future work with ICE will focus on the trade-offs between a single application approach and an approach where multiple applications are needed to access from multiple devices. The goal of implementing one application for access by many type devices is desirable, and preliminary data shows that his is feasible to some extent if adopting the information centric approach. The question that arises though is: when is the interaction style and presentation of a particular application different enough when presented on two device, that it no longer can be considered the same application. These differences belong to three categories, access to functionality, presentation of information and in the functions that are invoked as a result of interactions.

REFERENCES

[1] Sullivan, Terry ,(2001) Hilbert Problems in Visual Interface Development/Evaluation, *Proceedings of JDCL'01*, Roanoke, USA

[2] Roth, Steven , Lucas, Peter, et al , (1996) Visage: A User Interface Environment for Exploring Information, *Proceedings of. Information Visualization 1996*. San Francisco, USA

[3] Tristram, Claire,(2001) The Next Computer Interface, *MIT Technology Review*, issue December 2001

[4] Lucas, Peter,(2001) Pervasive Information Access and the Rise of Human-Information Interaction , *Presented at an Invited Session at CHI'01*, Seattle Washington. USA

[5] Schilit, Bill, Trevor, Jonathan et al, (2001) m-Links: An infrastructure for Very Small Internet Devices, *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking 2001*, Rome, Italy.

[6] Carreiro, Nicholas, Fertig, Scott et al (1994) The "Lifestreams" Approach to Reorganizing the Information World *Technical Report, Yale University* 1994

[7] Fertig, Scott, Freeman, Eric et al, (1996) "Lifestreams": An alternative to the Desktop Metaphor, *Presented as a Video at CHI'96*. Vancouver, Canada

[8] Gershon, Nahum, (1995) Human Information Interaction, *Presented at World Wide Web 4 conference*, December 1995, Boston, Massachusetts, USA

[9] Jonsson, I-M (1999) The Decoupled Application Interaction Model – DAIM. *Published in the proceedings of HCII' 99*, Munich, Germany.

[10] Hodjat, B., Amamiya, M., (2000), "Introducing the Adaptive Agent Oriented Software Architecture and Its Application in Natural Language User Interfaces", in *Proceedings of 1st international workshop on AOSE,* Limerick