Meta Design – End Users Creating Content

Final Project Report

Due Date: April 25th, 2005

Project Participants:
Ali Hassani
Javier Velez

Table of Contents:

1 Introduction:

Meta-Design could be considered one of the most interesting concepts in software design and development. It is interesting because nowadays everything has become mutable. Our environment is increasingly configurable which makes the configuration tools easier to understand and more common to use among people. The simplicity of these tools helped everyone acquire a better knowledge of the technical environment around them and made it simpler for experts to merge them into the development of more robust systems. Hence, concepts like Meta-Design and End-User development (EUD) activities have become a better way of thinking towards making systems that suit users' needs and satisfy the eager of developers to construct superior systems.

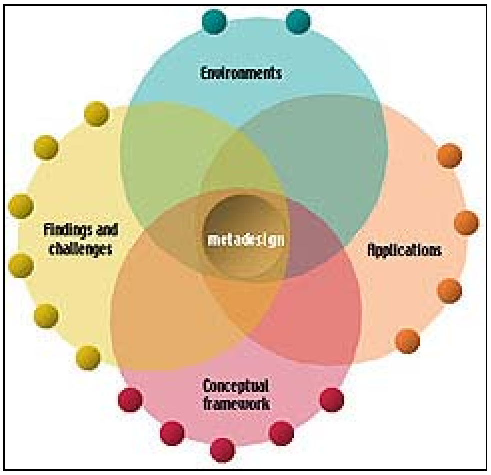

Figure 1: Meta-Design as common denominator

Figure 1 shows Meta-Design as a common factor between the usefulness of configuration applications, the complexity of findings and challenges, the organization of conceptual framework and the versatility of the development environments.

The new foundation of Human-Computer Interaction (HCI) systems would be in the biggest need for Meta-Design and social groups' collaboration in order to build every system. The new systems have to have a domain of shared ideas and inputs from concerned people in order to guarantee a better product, and collaboration could only improve the making of products because designers would be able to see the products not only from their point of view, but from other people's point of view as well.

2 Vision of Meta-Design:

In Meta-Design, configurations and techniques have the role for creating new tools to End-Users to see the problem as designers and become code-developers. Hence, its goal is to create a social technical environment to encourage users in the process of designing and making of systems rather than just be limited to the functionality that was provided to them.

Typically, the design process is divided into two main phases: a design phase and a use phase. The design phase is the phase in which the system gets assembled and built. The use phase is the phase in which the system gets used by the end-users. By including Meta-Design and all the concepts that come with it to the design process, we will add another phase that is characterized by the participation of the users' representatives to share ideas about what they expect from the system and about the input as to how to achieve that. The collaboration of the users' representatives is done through the technical tools provided by the developers in the phase called **under-design**. **Under-design** is a small phase of the design phase in which the system developers design tools and techniques to allow end-users to collaborate in the project to make sure that the product will suit user's needs and avoid major modifications to the system after it has been built. *Figure 2* shows the changes to a Meta-Design process.

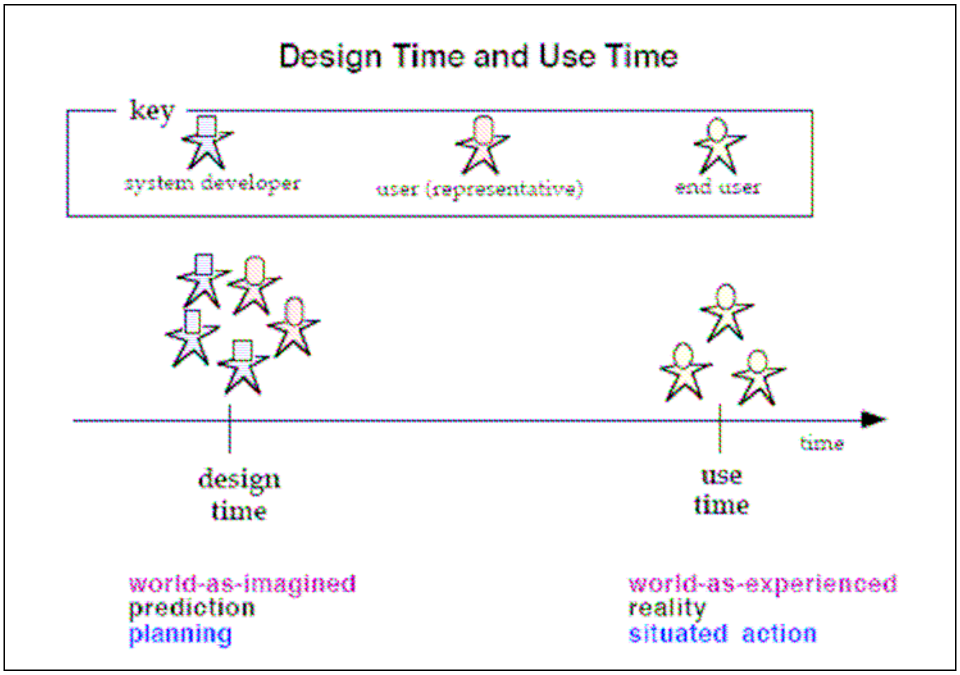

Figure 2: Design process with Meta-Design

3 Analyzing EDC and Clever Projects:

Envisionment and Discovery Collaboratory (EDC):

The EDC system is a prototype application to assist citizens of neighborhoods in constructing a transportation system that suits everyone's needs. This is a system that is built mainly by end-users for end-users. A group of developers has constructed a configuration tool to give users a real time picture of what a bus line would look like if it passes by the neighborhood. At that time, users have a better idea as to where to place a bus stop that would be convenient to them. Using 3D physical objects they can manipulate where the bus stop needs to be and what the trajectory of the bus should be. After several tries, the users would be able to build a consensus about what the trajectory of the line should look like and where the stops should be. *Figure 3* shows three people using the EDC prototype.

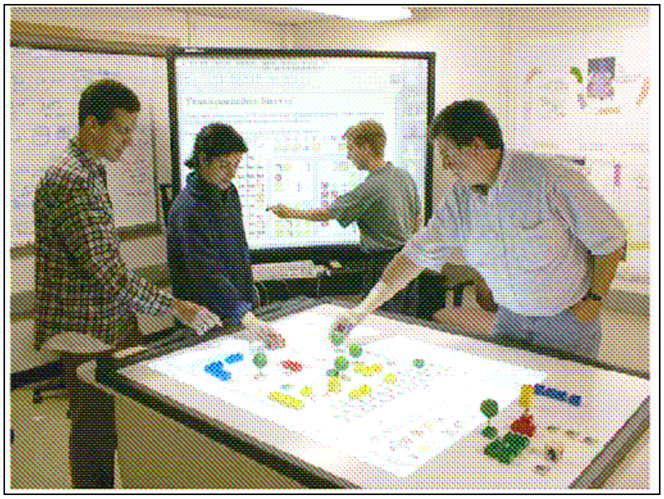

Figure 3: EDC Prototype

From all the aspects of the EDC prototype, we saw that a project like this would have never existed if there wasn't a true collaboration from the end users. It is a system designed for the end users by end users through their input. It is also a true implementation and application of the design and collaboration principles that the concept of meta-design is built on. The design and collaboration principles are applied to a system that is intended to be used to solve many problems. From taking a look at the EDC project, we can deduct that end users or stakeholders are the best source of requirements and specifications to most of the projects. We are past the era where we are designing operating systems and needing input from experts of the field rather than regular people. This era is characterized by applying all the new things that the advancement in technology has brought to us into our everyday lives, therefore user input and Meta-Design are very necessary components to make systems that satisfy users needs.

CLEVER:

The CLEVER project is a system that helps people with cognitive disabilities to use public transportation safely and properly. It is a system that serves as a bridge between the person that has a cognitive disability and the care giver of that person in order to ease the interaction between the two of them if the disabled person gets stuck or mixed up as to what bus to take or what is the way home. The CLEVER system includes all the data about the buses that the disabled person takes, it includes a Global Positioning System (GPS) to enable the care giver to track the location of the disabled relative and to make sure that he/she arrived to his/her destination safely. Hence, it is a powerful tool of interaction and data delivery to many people that suffer from cognitive disabilities. *Figure 4* shows a closer look to how the CLEVER system is supposed to perform its tasks and to get its data.

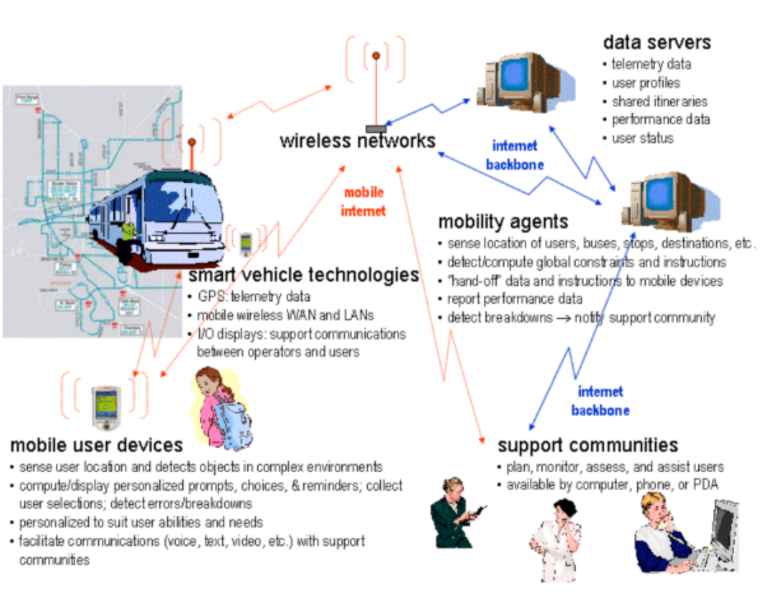
data servers
• telemetry data
• user profiles
• shared itineraries
• performance data
• user status
wireless networks
internet backbone
mobile internet
mobility agents
• sense location of users, buses, stops, destinations, etc.
• detect/compute global constraints and instructions
• "hand-off" data and instructions to mobile devices
• report performance data
• detect breakdowns → notify support community
smart vehicle technologies
• GPS: telemetry data
• mobile wireless WAN and LANs
• I/O displays: support communications between operators and users
internet backbone
mobile user devices
• sense user location and detects objects in complex environments
• compute/display personalized prompts, choices, & reminders; collect user selections; detect errors/breakdowns
• personalized to suit user abilities and needs
• facilitate communications (voice, text, video, etc.) with support communities
support communities
• plan, monitor, assess, and assist users
• available by computer, phone, or PDA

Figure 4: Concept of CLEVER

When looking at the CLEVER project, we can see that this is also another type of systems that could not be built unless there is a major contribution from the disabled person and his/her care giver. Specific information about the disabled person, about the bus routes ands the way home have to be provided in order for the CLEVER system to fulfill its task, and that's what Meta-Design appears to be very useful, because there is an important phase of the design of CLEVER that needed to happen and that was the direct involvement of the end-users in the making of it.

4 A closer look to Open Source: UNIX

The open source concept is a very wide concept. It assembles many problems and concepts related to issues of intellectual knowledge, software design, collaboration, leadership, and motivation. The collaborative issue is the one that has been studied while examining open source systems because the aim of this type of systems is to be able to create of a sociological and technical environment where participants can collaboratively construct solutions to problems of mutual interest. These ideas have been the catalyst to create new educational activities related to open source projects. The analysis of open source project could also be the first step in applying the open source principles in the making of other projects based on collaborative construction. For our analysis of open source systems we chose UNIX as an example.

UNIX is an operating system upon which the Internet was developed. It was developed in the late 1960s/early 1970s as a joint venture between General Electric, AT&T Bell Laboratories, and Massachusetts Institute for Technology. Pure UNIX is based upon a command line interface. However, just as DOS has Windows to provide a

GUI environment, UNIX has GUI overlays as well -- the two most notable are NextStep and X Windows. There are several free versions of UNIX, such as Linux and FreeBSD.

UNIX is an open source system that has proven to be much better than other operating systems and more adaptable to the end user because it is a creation of the users themselves. The main technical tool that UNIX offers to its users in order to add their input and help in its progress is the open source code (code always available to users) and the option of modifying it according to the needs of users.

UNIX Advantages over Other Operating Systems

For the last three decades UNIX has proven its superiority and stability compared to other Operating Systems. Here are some aspects of its superiority:

- UNIX is multitasking; therefore, multiple programs can run at one time.
- UNIX is multi-user, allowing more than a single user to work at any given time. This is accomplished by sharing processing time between each user and utilizing distributed computing systems.
- UNIX is safe; preventing one program from accessing memory or storage space allocated to another, and enables protection, requiring users to have permission to perform certain functions, i.e. accessing a directory, file, or disk drive.
- UNIX is more flexible and can be installed on many different types of machines (i.e. main-frame computers, supercomputers and micro-computers).
- UNIX is more stable, therefore requires less administration and maintenance.
- Finally, UNIX has greater built in security and permissions. These are some aspects of it.
    - It has a very clean separation between the kernel of the operating system and the rest of the applications. In windows, the boundary is blurred and has changed on several occasions.
    - The Windows registry is a black hole of security.
    - With the open source variants of Linux, it is easy to determine the version numbers of the software components: you just look at the code. With MS-Windows, frequently their own tools fail to detect critical software configuration changes.
    - UNIX has a thirty year history of dealing with security issues. A lot of what we know about the security of operating systems from the early 1970s is still relevant today. The UNIX security setup is completely transparent to the system administrator.

To sum it up, it is nice to know that most of the features that make the UNIX operating system stand out from the rest of operating systems are created by end users that are constantly involved in the its improvement through their input in the design and implementation. This is another example that shows that the employment of concepts like Meta-Design and open source could only be beneficial to a product by making it robust and widely used and understood.

5 Involving people in the making:

Although the implementation of Meta-Design seems straight forward as a concept, it seems that it is always hard to find motivated users and people to participate in the different phases of the design. Some people prefer to be happy using a product with limited functionality rather going and spending a long time trying to learn about certain product. The importance of the task that the user could play in the making of a product makes it necessary for the developers of the product to make an effort in simplifying the role of the user and, at the same time, not downgrade its importance. Therefore, the need of creating simple, fun and educative tools for end users to encourage them to participate in a project becomes great.

We found, from re-studying the EDC and CLEVER, that adopting the technique of "Role playing" could turn out to be a motivator for end users because it is a fun activity (everyone likes playing roles) and the goals that are deducted from it are very beneficial for the project and illustrate for the most part what the input of a general user will look like.

"Role playing" is very easy to perform and does not require much out of the users, all they have to do is study their roles well, and that will be enough for a first step in motivating users. The nice thing about that is that when users learn about their roles in a certain project, they are learning about the project in parallel, and if they come back to play an advanced role in the same project, then they will learn more and more as they go on and they will contribute more and more to the design through the continuous input that they are giving.

However, there is a lack of effort in making users motivated to get involved in any project. One of the efforts we could suggest is to involve meta-design techniques in engineering classes and include research projects into students' curriculums. Students of a certain area will always be trying to learn more and improve their skills in their field, therefore a class like Design Learning and Collaboration (DLC) in Computer Science would always be needed to expose students to projects and encourage them to be part of them.

6 Conclusions:

The creation of systems in which users can be designers is the main idea behind creating UNIX and systems like EDC and CLEVER. Our course for example is an environment where students are active contributors to their education through Swiki and Google..., and the envisionment and discovery collaboratory is an example of a system supporting meta-design (All of the projects Analyzed).

UNIX, EDC and CLEVER rely deeply on active participation, and that's what makes them good, changeable and reliable. For our case, contributing to an open source project like UNIX would be in the biggest educational benefit for us, and won't be hard to achieve. For everyone else, it is a great way to improve a product that they probably use and a great learning experience. Throughout this study, the benefit of how much people will learn after their contribution to an open source or a Meta-Design project, and how much back they could get out of it (in terms of their ideas being heard...) decides why people in different contexts become involved in open source and Meta-Design development.

7 References:

http://l3d.cs.colorado.edu/~giaccard/metadesign/

http://l3dswiki.cs.colorado.edu:3232/dlc-2005/uploads/9/meta-design-feb21.pdf

http://l3d.cs.colorado.edu/systems/EDC/

http://l3d.cs.colorado.edu/~gerhard/papers/tochi2000.pdf

www.netdictionary.com/u.html

http://linux.about.com/cs/linux101/a/unix_win.htm

http://l3d.cs.colorado.edu/~gerhard/papers/tochi-social-issues-final.pdf

https://blogs.msdn.com/steverowe/archive/2005/04/15/408688.aspx

http://www.upperhost.com/windows_unix_host.htm

http://hosting.nettro.com.au/plans/unix_vs_windows.htm

http://www.sg.gs/unix_vs_windows.htm

http://www.hyper-ad.com/unix/essentials/UNIX_vs_Others.html

http://www.bluehaze.com.au/unix/unixnt.html