# Requirements Elicitation for an Intelligent Software Test Environment for the Physically Challenged

**Warren Moseley Ph.D.**

St. Andrews Presbyterian College

Laurinburg, NC 28352, USA

910-277-5252

moseleycw@yahoo.com

## ABSTRACT

This paper is about the elicitation of the requirements for an intelligent interface for a software test development environment that will accommodate the physically challenged (PC). This research explores the use of eye-tracking mechanisms and digital manipulative user interfaces that are especially enhanced for the PC. In addition these devices provide assistance for the knowledge elicitation phase for an Intelligent User Interface to such an environment. It was never a stated objective of PCTA (Physically Challenged Test Assistant) to include any intelligent augmentation of the environment. It was challenge enough to get a paraplegic to operate the software test environment. However, in the process of evaluating the data collected in the evaluation of the user interface it was discovered that empirical data existed to predict some of the impasses that occur in the software development and more uniquely in the software testing process.

## Keywords

Knowledge Acquisition, Knowledge Elicitation, Scenario-Based Engineering, Software Architecture, Design Patterns, Physically Challenged, Eye Tracking, Digital Manipulatives, Object Oriented Architecture, Americans with Disabilities Act (ADA) of 1990, Intelligent Process Automation.

## INTRODUCTION

PCTA is a subset of a larger research effort called the Physically Challenged Assistant (PCA). PCTA involves studying the testing patterns of the Physically Challenged (PC) and the non-PC software developers to provide interface and coordination requirements necessary to build a software test environment and to aid the PC person to integrate into a multi-user coordinated software team.

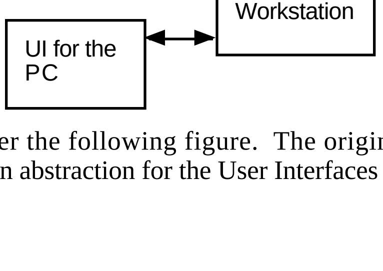
PCTA is a software architecture framework for understanding the system components and their interrelationships of a Software Test Environment to support the PC. This understanding is necessary for the analysis of existing systems and the synthesis of future systems that will contain knowledge intensive components.

The Americans with Disabilities Act (ADA) of 1990 stipulates that employers must make reasonable accommodations for those with disabilities. It is not easy to determine just how a physically challenged person will interact with an environment, and hence employers have focused their attention to jobs that require only one person to accomplish and usually ones that are not time critical. Seldom in the American or International workplace would you find a quadriplegic as a part of multi-user coordinated software development team. The Physically Challenged Assistant (PCA) covers situations that are time-dependent and ones that include user interaction and task coordination. PCA allows employers to more readily conform to the ADA, and to mainstream valuable assets.

## SOFTWARE FRAMEWORKS ENHANCE THE KNOWLEDGE ELICITATION AND SYSTEMS ANALYSIS PROCESS

In support of analysis and knowledge elicitation, software architecture frameworks capture domain knowledge and community consensus. They facilitate evaluation of design and implementation components. They ease simulation and prototyping of environments to support the target users. In support of synthesis, architecture frameworks provide a basis for establishing product lines and using domain knowledge to construct and maintain modules, subsystems, and systems in a predictable manner. PCTA provides this framework for reuse and understanding. An example of a software framework that has provided serious return on the initial investment is the ACE[ ] (Adaptive Communication Environment) and the Computer Integrated Manufacturing (CIM) Applications Framework developed at SEMATECH.[1]

The realization that given the extra emphasis on the user input, it was empirically possible to study the testing patterns of the PC and the non-PC software developers. This data is used to provide interface and coordination requirements necessary to build an intelligent software test environment to aid the PC person to integrate into a multi-user coordinated software team.



Consider the following figure. The original idea was to create an abstraction for the User Interfaces (UI) for the PC and the Non-PC. Implementations of these separate abstractions would provide a suitable working environment. We expected little difference between the workstation level and the actual end product. The latter has proven true, but the abstraction of the interfaces brought up some more significant issues. In examining the data from the original prototype the PC person provided more data than the non-PC person did, i.e. focus of attention. The focus of PC person attention could be the derived from eye tracker data. Focus of attention was never an objective of the original concept. The question arose, just what is the person thinking and how does that affect the approach to the software testing process. The user interface must support the role of a coordinated team player. This requirement differentiates PCA from just adding devices to an existing software support environment for the PC.

## A SIMPLE OVERVIEW OF SOFTWARE TESTING

A quality analysis-oriented test of a computer program consists of:

1. Running the program with a controlled set of inputs
2. Observing the run-time effect the inputs have on the program
3. Examining the program outputs to determine their acceptability

This may sound trivial for a non-PC person but without assistance just controlling a set of inputs, observing the run-time effects, and analyzing the output can be a monumental, sometimes impossible task for a PC individual. A quadriplegic just cannot crawl under a desk to connect a network interface cable, flip the switches on a density meter, or manipulate the dials and gauges of a water quality instrument. PCTA makes the interface for the PC as seamless as possibly feasible. Our original approach to evaluation of this environment was to question the user and managers to determine the ease and level of usage, but now focus of attention is non-intrusively traceable.

Contemporary testing technology advocates both a static and a dynamic quality analysis process. A dynamic test constitutes the process of running programs under control and observable circumstances. Static testing includes manual code inspections, structured walkthroughs, or the use of automated tools that analyze software by looking for certain kinds of common errors. Just holding a code inspection or structured walk-through can sometimes be a monumental task for the PC individual. PCTA attempts to combine the best of static and dynamic testing.

## CHALLENGES FOR THE PC

The computer is now, and will continue to be an essential part of the toolkit for the physically challenged. It offers new hope of extended independence to those who previously had little hope. These rapid technological advances translate to increases in system complexity. The ability to study and to manipulate complex computing environments is critical to the creation of working and meaningful environments for the physically challenged. An important part of the cooperative environment is the communication devices for the PC individual. Using these devices as a means to a solution creates a distributed application because these interface devices are in themselves powerful and complex computers with very sophisticated application software, some of which already operate as distributed systems. Following are some of the devices that make PCA a reality.

### The Liberator

The Liberator is a specially modified computer adapted to the needs of the PC.

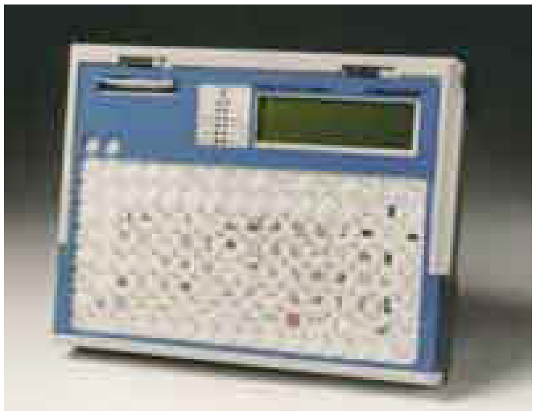


**Figure 1 - The Liberator**

The extended demands for multi-user coordination in PCA make it necessary to encapsulate the functionality of the Liberator into a larger distributed computational model. Liberator's capabilities can be extended to contain powerful programming tools to augment the communication capability of the PC.

Liberator’s clock and calendar serve as organizers that are particularly useful in the software development process and in research associated with the development process. In this study we used this device as one of the interfaces to the software test environment. But of greater significance a slight modification provides a non-intrusive monitoring and data acquisition mechanism to help determine just exactly what the PC person was doing. In the initial findings time was a factor that impacted the overall software development and hinders the PC participant from being an immediate asset to the development team.

Though it is possible to derive some of the cognitive task data from the Liberator interface, we found it limited when trying to track exactly what the participant was doing at a fine-grained micro level. For instance we could tell that the person was working on debugging a segment of code, but it was difficult to determine just what cognitive activity was activated. Coordination of chronological elements with other parts of the environment produced task sequences at a coarse grain level. Often a shift of attention to an associated document, diagram, or piece of test equipment alluded the Liberator. The Liberator allows for other assisting devices to be added to enhance and monitor the software development interface .

### The Eye Tracker

The ION™ Eye Controlled Cursor Control System is an access device that operates completely through head and eye motion giving the PC person total access to the computing environment.

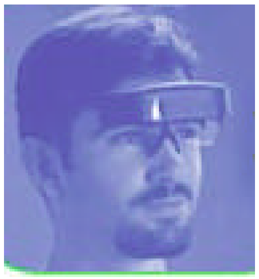


**Figure 5 - The ION™ Eye Tracking Mechanism**

Software adds eye control to basic software for head control, and allows full control of a computer with only the eyes. Two tiny cameras in the headset observe both the user's eye and the beacon on your monitor, allowing the computer and the ION™ Eye Control Software to determine where the PC Person is looking on or off the screen. The ION™ can also sense intentional blinking, and uses that for clicking and dragging. Eye tracking mechanisms give us insight into the cognitive tasks for PCTA. This will provide a focal point for future studies into the cognitive aspects of the software testing process.

### Digital Manipulatives for the PC Person

In many educational settings, manipulative materials (such as Cuisenaire Rods and Pattern Blocks) play an important role in learning and enabling the exploration of mathematical and scientific concepts through direct manipulation of physical objects**.** Resnick [2] and MIT researchers have developed a new generation of "digital Manipulatives" -- computationally enhanced versions of traditional children's toys have been developed by the Media Lab at MIT. These new manipulatives enable exploratory investigation of design concepts. PCTA uses digital manipulatives such as computationally augmented versions of blocks and tubes to help create a construction environment that are mentally challenging and physically useful to those who are physically challenged.

Resnick [3] sees the use of digital manipulatives as part of a broader trend within the computer interface research community. Manipulative objects have traditionally been abstract objects, such as those found in object-oriented languages and direct-manipulation graphical interfaces. Physical object analogs exist for most abstract objects.

In research efforts variously described as "ubiquitous computing," "computer-augmented environments," and "things that think," [4] researchers are now exploring ways of adding computational capabilities to everyday objects ranging from notepads, desktops, eyeglasses, new Liberators, and the ION™ Eye Control System.

Resnick and the MIT Media Lab researcher focus their attention to direct manipulative learning aspects, and in particular the use of the physical devices as a part of the learning environment for young children. PCTA switches the focus of their attention to the use of directive manipulatives as a means of adding functionality to the participatory software development environment. The use of these devices as a mechanism for non-intrusive means of data collection and knowledge elicitation for the PC individual provides a foundation for an adaptive development environment based on an intelligent user interface.

### Programmable LEGO™ Bricks

The MIT Programmable Brick is a tiny, portable computer embedded inside a LEGO™ brick, capable of interacting with the physical world through sensors and motors. PCA significantly reduces the volume of input necessary for the PC person by using visualization and direct digital manipulative techniques. PCTA was the first target area for digital manipulatives by PCA in a distributed object computing environment. In the first prototype we used the LEGO™ RCX Brick.

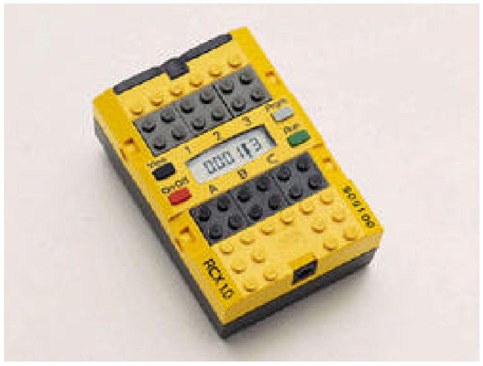


**Figure 5 the LEGO™ RCX Brick used in PCTA**

Several projects in a lake ecology study required extensive software enhancements to the scientific laboratory equipment software to include new real-time data acquisition capability. We chose a water quality experiment and the software test environment for the equipment used to monitor water quality to test our strategy of using direct manipulatives for the PC.

Figure 6 shows lab equipment with RCX bricks attached. Specially modified panels were placed in each of the operational modules of the laboratory test equipment. There were usually several RCX bricks and other interface apparatus that allowed computer controlled operation.

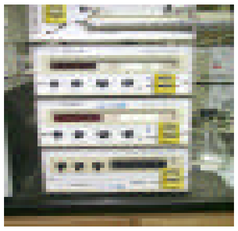


**Figure 6 - Lab Equipment with attached RCX Lego™ Bricks**

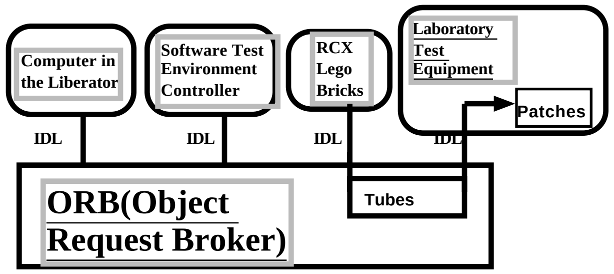
. We attached the infrared sensors and transmitters to designated portions of the laboratory test equipment and likewise modified the user interface devices to use the infrared sensors and transmitters of the LEGO™ Brick to control the test equipment hardware to make the manipulation of the software test environment hands-free. These devices can be operated by screen representation of the physical devices from computer interfaces, but also from embedded processors in devices such the ION™.

## THE PROGRAMMING ENVIRONMENT

In PCA wires are both a practical and conceptual nuisance. They limit not only what the user can build but also how they think about their constructions. The part of PCA that is unique is that it provides a new way to think about and explore software products, software processes, and more importantly the software design process for PC people. PCA gets rid of the wires and embeds computational capabilities directly in the digital manipulatives. The wires in LEGO™/Logo[5] confuse the kids, and in PCTA they are unnecessary physical hurdles for the PC individual. The PC person just can't crawl under a desk and connect a wire to the NIC(network interface card) of a computer, and then connect that wire to the computer in the lake ecology lab equipment. These scientific experiments were restricted to inside the lab, but the lake ecology project involved the installation of air and water temperature sensors in the lake. We did not attempt to have the PC install these outdoor sensors.

## PCTA USES A CORBA-BASED SOFTWARE ARCHITECTURE

PCTA uses a client server architecture model derived from the Object Management Group's (OMG) Common Object Request Broker (CORBA)[6]. The most important facet for selection of CORBA as a basis for PCTA was the potential of subsuming every other form of Client/Server Middleware. CORBA uses objects as the unifying metaphor for bringing existing applications to the bus. At the same time it provides a solid foundation for a component-based future and Software Framework Construction.



Computer in the Liberator
Software Test Environment Controller
RCX Lego Bricks
Laboratory Test Equipment
Patches
IDL
IDL
IDL
IDL
Tubes
ORB(Object Request Broker)

**Figure 8 - Corba Diagram of PCTA Architecture**

CORBA does not have a LOGO binding. Open/Doc does. It is possible to simulate a CORBA Object Request Broker (ORB) using Open/Doc there are computers in the Liberator and all of the rest of the interface devices to support the PC person. All of the software applications in all of the computers that are connected to PCTA are either Object-Oriented applications or have been wrapped to appear to use distributed objects. Figure 8 shows a highly simplified architecture diagram for the PCTA. The use of CORBA architectural patterns provides the foundation for the creation of a framework for a software test environment that is adaptable to the needs of the PC.

The PC Person no longer has to deal with the constraints of the physical environment. The use of the digital manipulatives provides an interface to the software architecture. The software architecture is able to be controlled by the devices and the computers in the software test.

## SUMMARY

This research involved the creation of a context for design and a support environment to create applications for the physically challenged (PC). It is extended to include scenarios in which the physically challenged will be a part of the development team. PCTA involves empirically studying the testing patterns of the PC and the non-PC software developer.

---

D. C. Schmidt, "An OO Encapsulation of Lightweight OS Concurrency Mechanisms in the ACE Toolkit," Tech. Rep. WUCS-95-31, Washington University, St. Louis, September 1995.

1 Computer Integrated Manufacturing Applications Framework Specification v 1.2, SEMATECH, Technology Transfer Document 93066196E-ENG, SEMATECH Consortium, Austin, TX. March, 1 1995

2 http://el.www.media.mit.edu/groups/el/papers/mres/black-box/proposal.html

3 Resnick, M., Bruckman, A., and Martin, F. (1996). Pianos Not Stereos: Creating Computational Construction Kits. Interactions, vol. 3, no. 6 (September/October 1996).

4 Resnick, M., Berg, R., Eisenberg, M., Turkle, S., and Martin, F. (1996). Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Instruments. Proposal to the National Science Foundation (project funded 1997-1999).

5 Resnick, M., Martin, F., Sargent, R., and Silverman, B. (1996). Programmable Bricks: Toys to Think With. IBM Systems Journal, vol. 35, no. 3-4, pp. 443-452.

6 R. Orfali, Client/Server Programming with Java and CORBA, John Wiley, New York, 1996, ISBN 0-471-16351-1.