ABSTRACT

INHERITANCE MODELS IN OBJECT-ORIENTED HARDWARE USING PHYSICAL OBJECT DEVICES

Vernon H. Mauery

Department of Computer Science

Master of Science

Several companies have attempted various ways of designing hardware devices that are simple to interface with and use. The toy industry wants to make electronic toys that kids and young adults can use, while do-it-yourself companies target the home electronics enthusiast who wants to interface with his electronic gadgets. The result has been classified as object-oriented hardware. This hardware offers features such as encapsulation, abstraction and inter-device communication, but it has not yet employed the concept of inheritance – one of the most powerful elements of software objects. Physical Object Devices (PODs) have a clean architecture that will allow us to easily extend the objects and find the requirements that inheritance places on its objects and environment.

By looking for parallels in hardware and software objects, we can employ inheritance models in hardware objects. These parallels will define the requirements for an

architecture using Physical Object Devices that has many of the same properties as software objects – including inheritance. Inheritance leads to the rapid development of more complex systems because object extension and interface inheritance offer code reuse and common interfaces. In addition to these basic time and money saving benefits, hardware inheritance will offer many of the properties that allow programmers to create high quality designs in software, such as encapsulation, member protection, and polymorphism. Developers who use inheritance with PODs will be able to create cleaner designs and more complex systems in less time and with less duplication of code.