멀티미디어 플레이어를 위한 RapidPLUS 기반 가상 프로토타이핑 설계 및 구현

임마누엘 아듀*, 안성순*, 이정배**, 최성희**
*선문대학교 전자계산학과
**선문대학교 컴퓨터공학과
e-mail : adufamel@gmail.com, ssAhn83@gmail.com, jblee@sunmoon.ac.kr, shchoi@sunmoon.ac.kr

The Design and Implementation of RapidPLUS Based Virtual Prototyping for Multimedia Player

Emmanuel Adu*, SungSoon Ahn*, Jeong B. Lee**, SungHee Choi**
*Dept. of Computer and Information Science, SunMoon University
**Division of Computer Engineering, SunMoon University

요 약

프로토타이핑 시스템 설계에서 프로토타이핑은 실제 모델과 같은 초기 모형을 뜻하는 것으로 개발 초기에 시스템의 모형을 간단히 만들어 사용자에게 제시하여 사용자로 하여금 실제 작동시켜 기능의 추가, 변경 내지 삭제를 요구하도록 하여 시스템을 점차적으로 개선시켜 나가도록 하는 방식을 말한다. 가상 프로토타이핑은 임베디드 소프트웨어 개발 초기단계에서 사용자 혹은 의뢰자의 요구사항들을 효과적으로 추출할 수 있도록 고안된 시스템 요구 및 제약조건 추출 방법론이다. 본 논문에서는 가상 프로토타이핑 기법을 이용하여 멀티미디어 플레이어의 다양한 기능을 시뮬레이션 할 수 있는 가상 프로토타입을 설계 및 구현한다.

1. INTRODUCTION

With the increased competition from the global economy, companies face the challenge of delivering new customized products faster to decrease time to market and meet customer demands. A delayed development or delivery may result in a business failure or in severe losses.

In most cases, a sample or prototype is often required as part of the design cycle, to allow demonstration, evaluation, or testing of the proposed product before the actual production begins. This fast process of creating a prototype is known as Rapid Prototyping (RP). Depending on the system used, the size and complexity of the prototype, creating a visual prototype with contemporary methods can take from several hours to several days. Therefore we can say that the word "rapid" is relative.

In spite of the fact that the term rapid prototyping is new, the use of prototypes as a way of testing ideas has a long and successful history. Asimow's (1962) Introduction to Design specifically mentions the use of prototypes as an empirical methodology [2]. Wilson and Wilson (1965) also describe prototyping as a design methodology [3].

The first techniques for rapid prototyping became available in the late 1980s and were used to produce models and prototype parts. Today, they are used for a much wider range. Many designers think it is advantageous to use virtual prototyping to design, visualize and test certain products digitally and evaluate different design concepts before making physical Prototype.

Rapid Prototyping is intended to reduce the time and cost of production, while increasing the flexibility and users involvement. The users can view a form of the final product, provide detailed feedback, and make constructive suggestions to achieve stability and reliability. In so doing, designers can identify and resolve issues and problems earlier in the process. As a result there is a reducing of the overall cost and time of production. We can therefore say that, the primary benefit of the Rapid Prototyping process is that it reduces the time needed to complete a project, thus reducing the expenses of the project by reducing the cycle-time of the design.

Lantz provides evidence that rapid prototyping pleases users, reduces development costs, decreases communication problems, lowers operations costs, slashes calendar time, and produces the right system for the designated task.[1]

Rapid prototyping requires the availability of tools (in most cases computer software) that offer modularity and plasticity. Modularity allows a portion of the design to be added, removed, or modified without affecting interactions in the other portions of the design as a whole. The second requirement, plasticity, refers to the ability to change some aspects of a design with only minor time or cost penalties. For example we can change the color of a prototype at wish without affecting the overall design.

Even though the development process may be very complicated, the product still needs to be released to the market quickly, in other words the designers cannot help but minimizing the development period. By taking these points

into consideration, a prototyping tools and methodology are needed to capture the business requirements to respond to the limited time, low cost, rapidly changing market, and needs, on time.

1.1 Basic prototype categories

In general, prototyping has four basic categories:

- **Proof-of-Principle Prototype (Model)**. This type of prototype is used to test some aspect of the design without trying to exactly simulate the visual appearance, choice of materials or manufacturing process. These types of models are often used to identify which design options will not work, or where further development and testing is needed.

- **Form Study Prototype (Model).** This type of prototype will allow designers to examine the basic size, look and feel of a product without simulating the functions of the product.

- **Visual Prototype (Model)** will show the intended design aesthetic and simulate the appearance, color and surface textures of the intended product but will not actually embody the function(s) of the final product.

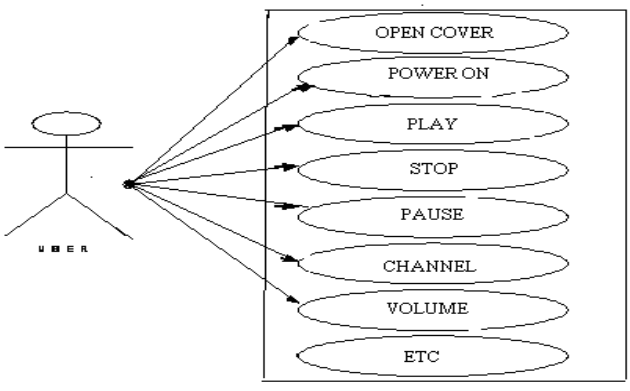
- **Functional Prototype (Model)** (which is a working prototype) will, attempt to simulate the final design, aesthetics, materials and functionality of the intended design.

1.2 Our objectives are:

1. To create virtual prototype for a multimedia player based on RapidPlus to simulate several functions of a multimedia player, such as power, volume, color, play, pause, stop etc [Figure 1]
2. To show transitions, behaviors, activities within the system.

The paper is organized as follows: Section 2 describes the system, and Section 3 presents the design of the models. In Section 4, we describe out our implementation and testing, and in Section 5 is our conclusion.

2. SYSTEM DESCRIPTION



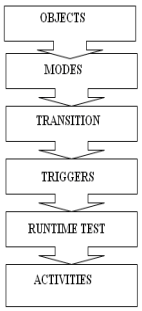
(Figure 1)-The use case of the media player

2.1 Designing the User Interface

There are two steps to design the user interface using RapidPLUS; arranging the object figures or locations and defining the property and motions of the objects. Thus, the designing stages can be represented detailed as Modes, Transitions, Triggers, Activities, and Runtime Test.

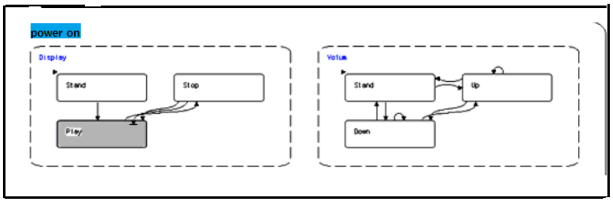
- Object: Designing the interfaces of the application program. Design the basic type of the object.
- Modes: A step to define the status of the available motion in the application program.
- Transitions: Transition between the application program modes.
- Triggers: Setting the conditions for events or transitions.
- Activities: Setting the motions of the mode
- Runtime Test: Test by using the prototype tool

[Fig 2].



(Figure 2)- Designing step for virtual prototyping

A Rapid application is a description of a complete system, such as our multimedia player. The system's overall behavior is broken down into individual **modes** [Figure 3]. For example, the player can be *on* or *off*. These are two distinct modes. It actually has many other modes besides *on* and *off*. While the player is on, there are several possible modes that you could describe: play, pause, stop, volume etc. The player, as a system, can be in one or several of these modes at any given time.

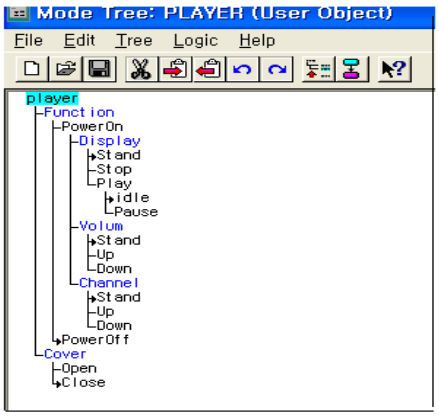


(Figure 3)- Modes decription

Each mode is a collection of activities which Rapid performs on the objects when the mode is active. Each activity can take place either upon entry into the mode, upon exit from the mode or for the entire time that the mode is active

3. SYSTEM DESIGN

When developing a Rapid application, we have to represent its behavior as a hierarchical arrangement of modes. The modes are linked as shown below [Figure 4



(Figure 4)- Mode hierarchy–tree form

The Rapid mode hierarchy is based on a parent-child relationship, with the arrangement of the tree branches depicting the relationship of one mode to another. The ***player*** mode breaks down into child modes, ***function*** and ***cover.*** We see that ***function*** is itself a parent to several children, referred to as sibling modes

Through this hierarchical organization we can see that when a transition is triggered from ***power off*** to ***power On***, Rapid deactivates ***power off*** and its children and activates ***power on*** and one or more of its children. For example when a transition is triggered from **power *off* to *power on, display, volume and channel*** (*which* are sibling nodes) will be activated with all their children.

3.1 Types of Modes

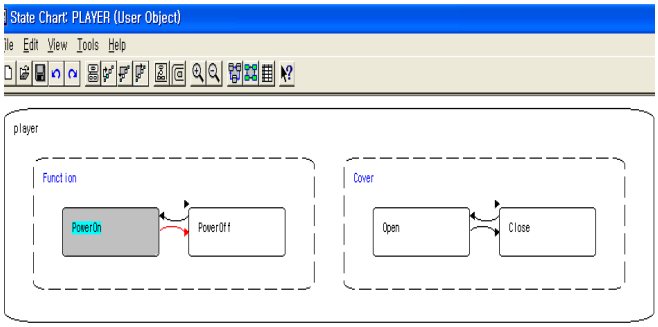
1) We call **power *on* and *power off*** exclusive modes, because only one of them can be active while their parent mode is active. Use exclusive modes for grouping activities that cannot run simultaneously

2) ***Display, volume*** and ***channel*** are called concurrent nodes because they are all active when their parent is active.

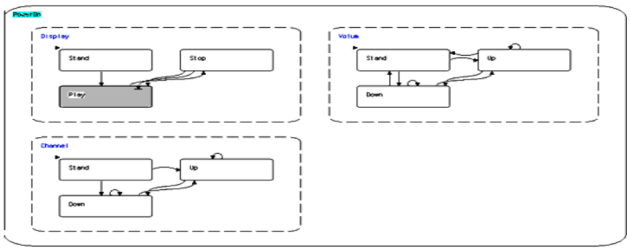
We use concurrent modes for parts of the system that are functionally independent of each other and can run simultaneously.

For example, our player can ***play*** a movie and change the ***volume*** at the same time.

Another way to represent the same hierarchy is in nested chart form. In this format, the hierarchy among modes is shown by nesting child modes within their parent mode [Figure 5] [Figure 6]



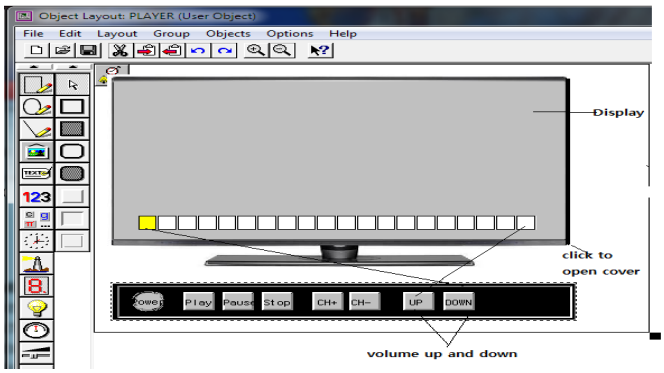
(Figure 5) Nesting child modes ***function*** and ***cover*** within ***player*** which is their parent mode.



(Figure6) -Nesting child modes ***display***, ***volume*** and ***channel*** within ***power on*** which is their parent mode

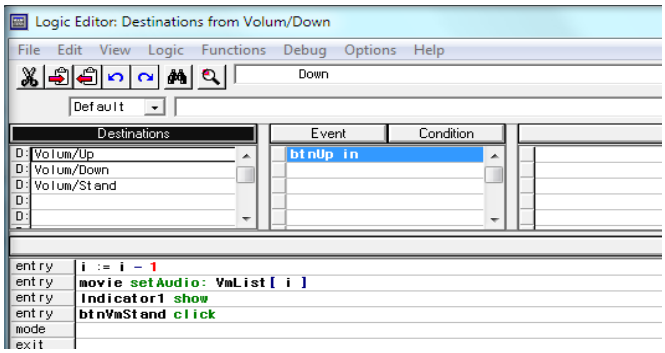
3.2 Creation of the Interface Module

The player can be designed by arranging the objects in the Object Layout window which is supported in the development tool for user interface modeling [Figure 7]



(Figure 7)- The Object Layout of the multimedia player

The Object Layout is a comprehensive tool for creating a visual representation of an application's layout. You use the Object Layout to add and define the graphic and nongraphic elements that comprise the application that you are developing.



(Figure 8)-Logic editor showing event of the volume node

[Figure 8] shows the logic editor which check each event that the simulation sends (for example the entries, modes, event and conditions of each activity is displayed orderly). If the simulation manager detects that the wrong event has arrived, it will trigger a transition.

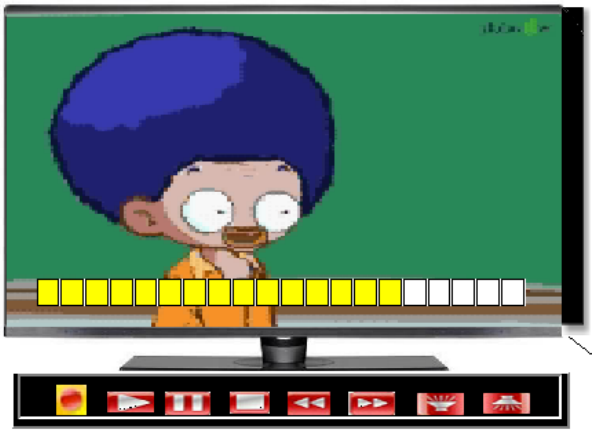
4. SYSTEM IMPLEMENTATION AND TESTING

Through the prototyper window of the RapidPlus, we can simulate various functions of our player [figure 9]. The user starts by clicking to open the cover. The cover then shows the functions of the player. When clicked, the power button

activates the functions of the system. A said previously, when the ***power on*** transition is triggered**,** the ***display, volume and channel*** *which* are concurrent nodes will be activated.

With the channel button we can select the file we want to play. At this stage we can manipulate the system at wish, namely by selecting a new file, increasing or decreasing the volume, by using the pause or stop functions or by turning the player off with the power off button.

When a transition occurs form play to pause the system becomes idle form the current position, so the user can resume from the current position by making a transition for pause to play. When a transition is triggered form play to stop the system becomes idle form the current position. The system automatically restarts to play the file form the beginning if the user triggers a transition from stop to play. Lastly when the user trigger the transition from power on to power off, except the power on transition all other transitions will be deactivated (or will not occur). We can see that, the system behaves differently base on each transition made to it.



(Figure 9)-Virtual simulation of the multimedia player.

5. CONCLUSION

In this paper we have tried to explain the necessity to use prototyping to reduce the time and cost of production. User's constructive suggestions and detailed feedback in the development process will help to resolve issues and problems, thereby achieving stability and reliability.

In addition using the rapidPLUS development tool, we've created a virtual multimedia player. We were able to virtually simulate each function of the player successfully.

참고문헌

[1] Lantz, K. E. (1985). *The prototyping methodology.* Englewood Cliffs, NJ: Prentice Hall.

[2] Asimow, M. (1962). *Introduction to design.* Englewood Cliffs, NJ: Prentice-Hall.

[3] Wilson, I. G., & Wilson, M. E. (1965). *Information, computers and system design.* New York: John Wiley & Sons.

<http://www.answers.com/topic/prototype>(05 March 2010)

[4] Rapid Plus, *Rapid Start* (2004), e-SIM Ltd.

Rapid Plus, *Rapid User Manual (*1997), Emultek Ltd.