

---

# 개선된 시각화 단축아이콘 관리자 모델 개발

김동현 · 김장수 · 정은혜 · 권오준

동의대학교 컴퓨터소프트웨어공학과

## Development of an Improved Model of Shortcut Manager with Visualization

Dong-hyun Kim · Jang-su Kim · Eun-hye Jeong · Oh-jun Kwon

Dept. Computer Software Engineering, Dong-eui University

E-mail : pairwork@nate.com

### 요 약

현대 사회에서 시각적인 요소는 제품의 기능 못지않게 중요한 요소로 작용하여, 시각적 요소에서 제품의 성패가 판가름 나는 경우도 있다. 하지만 이러한 상황 속에서 제품의 기능은 배제되고 시각적 요소만을 개발되는 경우가 발생되고, 겉보기에 화려한 제품들이 사용자들의 외면을 받기도 한다.

본 논문에서는 사용자의 외면을 받은 응용 사례를 분석하고 평가하여, 유용성을 개선한 시각화 단축 아이콘 관리자모델을 제안하고 개발하였다.

### 키워드

단축아이콘, 바탕화면, 아이콘, 시각화

### 1. 서 론

시각적인 효과는 보는 이에게 보는 즐거움뿐만 아니라 효과적인 전달력을 제공한다. 이러한 전달력은 짧은 시간 내에 강한 impact를 남길 수 있는 가장 좋은 방법 중 하나이다. 우리가 흔히 알고 있는 Windows, Mac OS, Linux 등의 Operating System 에서도 첫인상을 남겨줄 User Interface 에 시각적 느낌과 효과에 상당한 노력을 투자하는 이유도 이러한 이유에서 일 것이다.

시각화는 시각적인 안정감을 제공하는 것에 그치지 않고, 여러 가지 장점을 가지고 있다. 같은 정보를 text로만 전달하는 것, voice 로만 전달하는 것 보다 시각화된 화면을 함께 제공하면 상상 이상의 장점을 가진다.

첫째, 같은 시간 내에 보다 많은 정보를 제공할 수 있다. text나 voice는 조금만 잘못 읽거나, 잘못 읽으면 본래의 뜻이 완전히 달라 질 위험이 존재하고, 비교적 읽거나 들어야 할 것들이 많으므로 같은 단위 시간 내에 전달 할 수 있는 정보의 양이 적다.

둘째, 같은 정보량이라면 보다 빠른 시간에 전달 할 수 있게 한다. 같은 양의 정보를 전달하게 되는 것 또한 첫째와 같은 이유를 들 수 있다.

셋째, 보다 쉽게 주의를 집중 시킨다. text나

voice에 집중하는 것 보다는, 비교적 형태가 간단하고 색채가 다양한 시각화된 정보에 대하여 집중하는 것이 일반적으로 수월하다.

이뿐만 아니라 시각화로 인해 얻어지는 장점은 매우 다양하며, 이러한 시각화에 의한 장점을 유발하는 것 또한 어렵지 않으므로, 비용: 효과가 우수하다 할 수 있다. 따라서 시각화를 단순한 디자인 요소로만 인식 할 것이 아니라 관심을 가지고 개발하여야 할 것이다.

하지만 시각화에만 치중하다 보면, 금세 다시 사용자의 외면을 받게 될 것이다. 시각화와 더불어 유용성이 있어야 한다. Apple의 Mac은 차별화된 디자인으로 마니아층을 형성하고, 주목을 받을 수는 있었지만, application 부족으로 편리한 컴퓨팅 환경을 보장 받지 못해 고전한 경험이 있다.

2장에서는 기존의 응용 사례를 소개하고 분석하여 평가하고, 3장에서 새로운 모델을 제안해 나간다. 이어 4장에서는 이와 관련된 기술에 대하여 설명하고 5장에서 application 의 API를 살펴보고록 한다. 마지막 6장에서는 결론을 통해 논문을 마무리 하도록 한다.

## II. 기존 사례에 대한 분석

기존의 사례는 Object Dock 과, Rocket Dock 두 가지가 있다. 두 가지 모두 freeware로, 무료로 설치하여 이용할 수 있다. 두 응용 사례를 살펴보고, 평가해 보도록 하겠다.

StarDock Corporation(www.StarDock.com)에서 개발한 ObjectDock 은 Mac OS의 화려한 시각적 효과를 Windows 운영체제에서 사용할 수 있도록 제작되었다. 다음으로 RocketDock은 RocketDock (www.RocketDock.com) 에서 다운로드 받을 수 있다. 전체적인 모양새와, 기능 등은 거의 동일하다.



그림 1. ObjectDock (1)



그림 2. ObjectDock (2)



그림 3. RocketDock (1)



그림 4. RocketDock (2)

위의 그림에서 보이는 것과 같이 아이콘을 application 에 위치시키고 언제 어디서나 손쉽게 실행할 수 있는 root를 제공한다. MouseOver일 때는, 아이콘의 size가 달라지거나 onClick 이벤트에도 효과를 적용하여 단순한 기능에 시각효과를 부여하여 dynamic 한 느낌을 준다.



그림 5. Object Dock & Rocket Dock's Setting window

Windows XP에서 제공하는 시작메뉴 측면에 위치한 빠른 실행과 기능상 다를 것 없는 프로그램이지만, 다양한 시각적 효과와 이에 대한 설정을 사용자가 스스로 설정할 수 있게 되어있다.

앞서 소개한 application 은 이전에 볼 수 없었던 화려한 시각효과를 적용시키고, 이 또한 사용자의 취향에 맞게 설정이 가능하도록 제작되었지만 정작 사용하는 사람은 드물다. 그 이유는 시각적 효과 외에는 우리에게 유익한 컴퓨팅을 제공하는 부분이 많지 않기 때문이다. Windows XP에서 제공하는 빠른 실행과 대비 했을 때 시각효과가 적용된 것 이외에는 특별한 것이 없다. 오히려 Object Dock은 1.85MB, Rocket Dock은 2.3MB에 달하는 높은 메모리 사용량을 보였으며, 이는 단축아이콘 관리자의 목적으로 사용하기엔 너무 무겁다는 평가이다.

## III. 작업 영역 전환 기술

Windows는 추가적인 보안을 제공하기 위해 Windows Station , Windows Desktop 아키텍처를 사용한다.

"Desktop"은 논리적인 화면의 표시로서, User Interface Objects (GDI Objects) 와 User Objects 를 포함한다. "Window Station"은 secure kernel object로 clipboard, atom table, 한 개 이상의 desktop objects를 포함한다. desktop object에는 대화형 window station 이고 default window

station인 "WinSta0" 이 존재한다. WinSta0 station은 User Interface를 표시하거나 User Input을 받을 수 있는 유일한 대화형 window 이다. 다른 모든 window station은 비대화형이다. 기본적으로, 대화형 window station에는 default, WinLogon, ScreenSaver의 세 가지 Desktop이 존재한다.

Desktop 역시 secure kernel object 이다. 새 desktop이 생성되면, 이것은 프로세스 호출의 window station과 연관되어있다. 그곳에는 항상 단 하나의 desktop만이 보일 것이며(대화형) 사용자의 input을 받을 준비가 되어있을 것이다. 또한 제공된 desktop은 대화형 window station I.e.WinSta0과 연관되어있다. 이러한 대화형 desktop을 Input Desktop 이라 한다.

window station에는 default, WinLogon, ScreenSaver의 세 가지 desktop이 존재한다.

default는 어떤 사용자가 처음으로 로그인 했을 때 만들어지고 보이는 desktop이다. WinLogon은 Windows의 처음에 로그인 할 때 보이는 desktop이며, ScreenSaver는 화면보호기가 작동될 때 보이는 desktop 이다. 이러한 desktop은 대소문자를 구분하지 않으며 이름이 중복될 수 없다. 하지만 다른 window station에 존재할 때는 중복이 가능하다. Window Station과 Desktop Objects의 숨은 목적은 프로세스간 최고수준의 독립성을 제공하기 위해서이다.

#### IV. MiniDock 모델 구현

MiniDock 은 시각효과 이외에 메모리와 CPU의 사용량을 줄이고, 유용한 추가기능을 제공함으로써 프로그램의 필요성을 높이고자 하였다. 시각효과는 메모리를 많이 필요로 하는 zooming 대신 color - gray scale 전환을 적용하였다. 부가적인 setting 기능을 없애고 시각효과를 simple effect로 적용함으로써 메모리 사용량을 1.78MB로 줄였다.



그림 6. MiniDock (1)



그림 7. MiniDock (2)

MiniDock개발의 주요 관심은 유용성, 즉 활용도를 높이는 데 있다. MiniDock을 사용하게 될 주 사용자층은 컴퓨터를 아주 단순 용도로 사용하지 않고, 컴퓨터 사용 환경에 어느 정도 관심이 있는 사용자층이라는 판단 하에, 아이콘 관리자 외에도

유용한 기능을 선별하여 함께 탑재하였다.

추가기능은 Ubuntu에 적용된 화면전환 기능을 탑재하였다. 화면전환 기능은 desktop 공간을 논리적으로 확장하여 마치 듀얼 모니터를 사용하는 듯한 컴퓨팅 환경을 제공한다. MiniDock 실행과 함께 실행되며, 화면전환은 Ctrl + Alt + 숫자 (1,2,3,4) 로 전환이 가능하다.



그림 8. 작업 영역 전환

화면전환을 위해 다음의 API와 클래스를 정의하고 사용한다.

GetCurrentWindowStation : 프로세스 호출의 current window station 핸들을 검색한다. 검색된 핸들은 window station의 모든 desktop을 열거하는데 사용된다.

CreateDesktop : 새로운 desktop을 생성한다.

OpenDesktop : 특정한 desktop을 연다.

CloseDesktop : desktop에 열려 있는 핸들을 닫는다.

SwitchDesktop : 특정한 desktop을 보이게 만들고 user input을 받을 준비상태를 만든다.(활성화 상태를 만든다)

GetThreadDesktop : desktop에 지정된 스레드와 관련된 핸들을 검색한다.

GetObjectInformation : window station 및 desktop object에 대한 정보를 제공한다.

CDesktopManager라는 클래스를 이용해서, desktop 관련된 활동들을 통합해준다.

#### VI. 결 론

설계자와 개발자의 간격이 좁아지면서 기능 만 큼이나 디자인 또한 중요해 지고 있다. 하지만 단순히 시각적 요소에만 신경 쓴다면 결코 발전이 없다. 기존의 훌륭한 디자인 요소를 갖춘 응용프로그램이 대중화 되지 못하고 사용자에게 외면을 받은 이유는 그 이상의 어떠한 무언가가 없었기 때문이다. 소형 응용 프로그램은 목적에 충실하여 기본적인 기능을 잘 수행하고 다른 응용프로그램과 차별화 되어야 한다.

MiniDock은 응용프로그램의 특성상 사용자가 컴퓨팅을 하는 동안에 항상 실행될 수 있다는 가능성에 무게를 두어, 사용자가 컴퓨팅 하는 동안 꾸준히 사용할 수 있는 기능을 응용프로그램에 탑재하고, 기존의 응용프로그램 보다 CPU와 메모

리의 사용량을 줄이는데 목적을 두고 개발하였다. 그 결과, Virtual desktop manager를 탑재하는 과정에서 메모리 사용량이 올라갔지만 기능 : 용량의 비율이 예상보다 낮게 나왔으며, 요즘 같은 고사양 PC시대에는 항상 켜 놓고 사용할 수 있는 가벼운 application이다.

이 외에도 화면 색상추출기나 PC 예약종료 등의 가벼우면서도 유용하게 사용할 수 있는 여러 가지 기능을 탑재하여 통합 응용프로그램으로 개발된다면 더 유용한 응용프로그램이 되지 않을까 생각해본다.

#### 참고문헌

- [1] <http://www.StarDock.com>
- [2] <http://www.RocketDock.com>
- [1] Sung-hyun Han, " WIN 32 API Programming ", 8. 31. 2002
- [2] Nikolai. M Joshuties, "C++ Standard Library Tutorial References", 3. 9. 2007
- [3] <http://blogs.msdn.com/coreinternals/archive/2009/08/19/session-window-station-desktop.aspx>
- [4] <http://blog.hani.co.kr/jeongmoonoh/>