

# 효율적인 화면요소 배치에 관한 연구

김태우<sup>†</sup>, 박선이<sup>††</sup>, 여정모<sup>†††</sup>

## A Study on the Efficient Layout Method of the Screen Components

Tae Woo Kim<sup>†</sup>, Sun Yi Park<sup>††</sup>, Jeong Mo Yeo<sup>†††</sup>

### ABSTRACT

The use of information systems improves a company's service quality and provides efficient information management. The establishment of an information system has become an essential element. In designing and developing a large number of screens when building an information system, As the number of screens increases, so does the cost and time. In addition, modifications can increase the cost of maintenance and exceed the cost of development. In order to solve these problems, there have been studies to design the screen simply and easily, but there is a shortage in reducing the maintenance cost. In this paper, we have studied the layout algorithm for managing the attributes of the screen components as data structures based on the designed screen and arranging the components on the screen. And we compared the proposed method and the methods that have been studied in the related studies. Screen component layout methods can automatically deploy screen components to consistently develop a large number of screens and reduce cost and manpower by reducing the time it takes to deploy screens.

**Key words:** Screen Component, Screen Layout, User Interface Layout, Screen Component Structure

### 1. 서 론

정보통신기술의 발달로 기업은 다양한 업무를 정보시스템을 이용하여 고객에게 간편하고 빠른 서비스를 제공하고, 많은 정보를 효율적으로 관리할 수 있게 되었다. 정보시스템이 기업 운영에 필수적으로 활용되도록 효율적인 개발을 위해 정보시스템 구축 방법에 관한 많은 연구가 이루어졌다[1,2]. 정보시스템의 구축은 크게 설계와 개발 두 가지 단계로 이루어진다.

애플리케이션 설계 단계에서는 시스템의 규모에

따라 수많은 화면을 생성하게 되고, 설계된 화면이 고객의 요구사항을 정확하게 반영했는지 검토가 필요하다. 많은 화면을 검토하고 수정사항을 반영하기 위해서는 화면을 생성하는 도구[3] 필요하며, 이를 위한 도구로 그래픽 기반 도구와 모델 기반 도구에 대한 연구가 이루어졌다. 연구된 도구들을 활용하여 많은 화면을 빠르게 생성, 검토, 수정을 수행하지만 프로그래밍 언어 및 애플리케이션 유형(모바일, 인터넷 프로그램, 윈도우 프로그램)에 따라 도구의 사용이 제한적이다. 또한 설계된 화면을 설계자와 개발자간에 공유하여 사용함에 있어 불편한 부분이 존재

※ Corresponding Author : Jeong Mo Yeo, Address: (48513) Yongso-ro 45, Nam-gu, Busan, Korea, TEL : +82-10-2859-0488, E-mail : yeo@pknu.ac.kr

Receipt date : July 22, 2019, Revision date : Dec. 27, 2019  
Approval date : Jan. 13, 2020

<sup>†</sup> Dept. of Computer Science Eng., Graduate School, Pukyong National University  
(E-mail : mtkim7895@pukyong.ac.kr)

<sup>††</sup> Dept. of Computer Science Eng., Pukyong National University (E-mail : psunyi@pknu.ac.kr)

<sup>†††</sup> Dept. of Computer Science Eng., Pukyong National University

※ This work was supported by a Research Grant of Pukyong National University(2019)

한다.

이러한 문제를 해결하기 위해 본 연구에서는 설계 단계에서 설계한 화면을 검토 후 개발 단계에 적용할 수 있고, 애플리케이션 유형이나 프로그래밍 언어에 제한받지 않고 사용할 수 있도록 화면의 구조를 설계하고 이를 데이터로 추상화하여 관리하는 방법을 제안하고자 한다.

## 2. 관련연구

애플리케이션 구축에서 처음부터 설계 및 개발을 완벽하게 완료하는 것은 매우 어려운 일이고, 구축 과정 중은 물론 개발이 완료된 이후에 수정이 필요한 경우가 많이 발생한다. 설계 단계에서 수정이 발생하면 상대적으로 적은 비용과 시간이 소요되지만, 개발이 완료된 이후에는 수정 시 더 많은 유지보수 비용이 발생한다. 개발과 유지보수에 소요되는 비용을 감소시키고자 자동화 설계 방법에 많은 연구가 이루어졌다. 그중 본 절에서는 화면 자동화 설계 방법으로 그래픽 기반, 모델 기반 설계 방법에 대해 다루고자 한다.

그래픽기반 화면 자동화 설계 방법은 개략적으로 스케치 된 화면의 모습을 바탕으로 개발자가 시각적인 도구[4-6]를 활용하여 UI(User Interface)를 직관적이고 간편하게 구현하는 방법을 제공한다. 그래픽기반 화면 자동화 설계 방식에서는 각각의 프로그래밍 도구에서 제공하는 기본 프레임워크를 기반으로 화면 표현 요소들을 위젯 형태로 제공하고, 사용자는 위젯을 원하는 위치에 마우스로 끌어다 놓으면 프로그램 도구가 해당 위젯의 정적 코드를 생성한다. 그래픽기반과 모델 기반 설계 방법은 제공되는 팔레트를 이용하여 직관적이고 쉽게 화면의 모습을 표현함과 동시에 개발자가 작성해야 하는 코드 일부분을 자동으로 생성하여 개발 시간을 단축시킬 수 있지만, 수동으로 코드를 편집하거나 속성을 세분화하여 관리하는 경우 개발자 간에 통일성을 저하하는 문제도 발생할 수 있으며, 수정사항의 검토가 개발 단계에서 이루어져 유지보수 비용을 줄이는데 어려움이 존재한다. 또한 화면을 생성하는 도구에 종속되어 프로그램이 개발되기 때문에 화면설계 도구 기반에서 개발이 이루어지는 제약사항이 존재한다.

모델기반 화면 자동화 설계 방법은 프로그램에 독립적으로 다양한 플랫폼에 적용할 수 있는 모델을

정의하고, 이를 기반으로 화면을 자동으로 생성한다[4,7-11]. 모델기반 화면 자동화 설계 방법은 요구사항 분석단계에서 도출된 다양한 모델을 기반으로 화면을 구성[12-15]하기 때문에 특정 프로그램에 종속되지 않고 화면을 설계하고 구현할 수 있다. 하지만 프로그램 도구와 시스템의 분류가 발전함에 따라 많은 모델들을 도출하고, 다양한 모델의 조합으로 화면을 구성하기 때문에 분석이 쉽지 않다는 점이 존재한다. 도출된 도메인 모델에 CRUD(Create, Read, Update, Delete) 동작을 적용하여 화면이 수행하는 동작을 기준으로 화면의 모습을 추상적으로 도출한다[16,17]. 또는 클래스 다이어그램, Task 모델을 기반으로 해당 화면에 필요한 입/출력 값을 수행할 수 있는 화면 요소들을 정의하여 화면을 생성하고 화면들의 구조를 관리한다[5]. 하지만 모델기반 도구의 경우 구성된 화면의 모습을 개발단계에서 확인할 수 있어 정확하게 설계된 내용을 확인하기 어렵다. 그래서 그래픽기반 도구의 위젯을 활용하여 모델기반으로 화면을 구성한다. 위젯의 활용으로 화면 생성자의 사용이 용이하고, 직관적으로 화면의 모습을 추상화하여 볼 수 있다.

하지만 화면 생성과 개발의 일관성과 편의성을 확보하여 설계와 개발간의 오류를 줄이고 시간을 절약하기 위해 생성된 화면의 소스코드를 자동으로 생성할 경우 그래픽기반 도구와 마찬가지로 개발 도구의 선택에 제약사항이 존재하게 된다.

## 3. 화면요소 배치 방법

기존의 방법론과 UI생성 도구들은 전사적인 시스템관점에서 분석, 설계, 개발이 이루어 졌다. 그렇기 때문에 기존에는 만들고자하는 애플리케이션에 중점을 두어 개발이 이루어지는 경향이 존재해왔다. 본 연구에서는 ITA/EA(Information Architecture/Enterprise Architecture) 구조[18]에 따라 업무, 데이터베이스, 애플리케이션, 기술 영역으로 구분하고, 영역별로 설계 및 개발 절차 중 애플리케이션 설계영역에서 화면이 설계[19-21]된 가정에서 화면요소 배치를 진행한다. 그리고 Java 기반의 윈도우 프로그램 기준으로 화면요소를 데이터로 추상화 하여 화면을 구성하고 추후에 프로그램 언어와 애플리케이션 유형별로 확장해 나가고자 한다.

버튼과 텍스트필드와 같이 화면을 구성하는 요소

들은 프로그래밍 도구나 애플리케이션 유형에 관계 없이 고유한 요소들이며, 화면을 구성하는 요소들의 속성은 이름, 크기, 위치 등의 기본적인 속성과, 프로그래밍 도구 및 애플리케이션 유형별로 디자인적인 부분에 의해 관리되는 추가적인 속성으로 구분 된다. 본 연구에서는 화면을 구성하는 요소들의 기본적인 속성을 기준으로 화면 구성요소 데이터를 추상화하여 데이터 모델로 관리한다.

Fig. 1은 화면요소들을 데이터 구조로 Java의 AWT 기반의 경우로 설계되었으며 추가로 다양한 프로그램 도구의 경우를 포함한다. 화면을 구성하는 요소들의 정보 중 해당 요소를 표현하는 정보는 프로그램 도구마다 동일하게 표현되기 때문에 화면요소는 하나의 객체로서 데이터를 관리한다. 해당 객체에서는 화면요소를 표현하는 기본정보와 화면에서 배치되는 위치정보를 우선적으로 관리하며 해당 정보는 프로그램 도구마다 동일하게 필수적으로 정보를 사용자가 입력해주어야 하는 값이므로 필수속성으로 관리된다. 기본정보 외에 색상, 길이, 굵기와 같이 추가적인 정보는 사용자가 입력해주지 않더라도 값이 제공되는 선택적 입력 속성이지만, 입력이 없는

경우 프로그램에서 제공하는 기본 값을 사용하므로 이러한 정보들을 필수속성으로 관리한다. 화면요소의 종류는 모든 프로그램 도구에서 사용하는 요소들을 객체로 정보를 관리하며, ComboBox나 Table과 같이 객체를 이용하는 데 있어 추가로 필요한 정보는 자식으로 관계를 맺어주어 관리한다. 또한 하나의 화면에서 프레임, 다이얼로그와 같은 요소는 다른 화면 요소들을 배치할 수 있는 판과 같은 역할을 수행하기 때문에 필수적으로 필요하며, 이들을 묶어 Container로 명명하여 서브 타입으로 구분 지어준다. Container는 다양하게 존재하며 추가되는 경우 서브 타입으로 추가하여 관리한다.

설계된 하나의 화면에는 매우 다양한 화면 요소들이 배치되고, 배치된 소속정보들을 관계정보로써 관리한다. Container에 소속된 관계정보는 CotainerInf 테이블에서 소속된 요소의 개수를 관리하고, 해당 요소를 찾아가기 위한 상세한 정보는 ComponentConf 테이블에서 관리한다. Panel과 같은 화면요소들은 최상위 컨테이너는 아니지만, 화면요소를 소속시키기 때문에 Container와 같이 관계 테이블을 관리한다.

Fig. 2는 ITA/EA구조에 따라 애플리케이션 설계

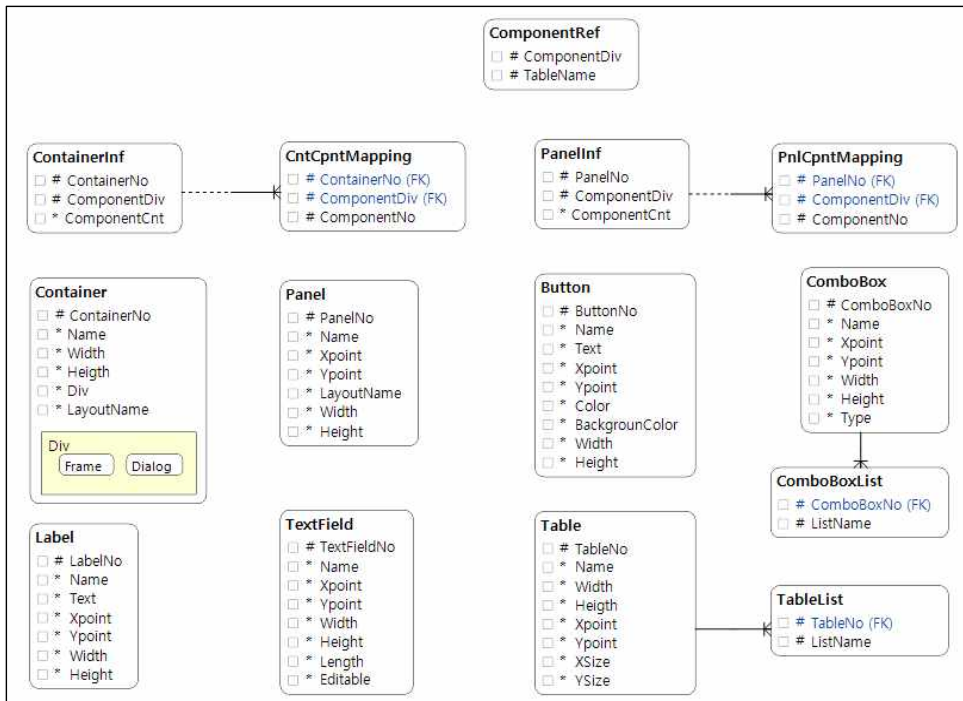


Fig. 1. Part of the Structure of Screen Component.

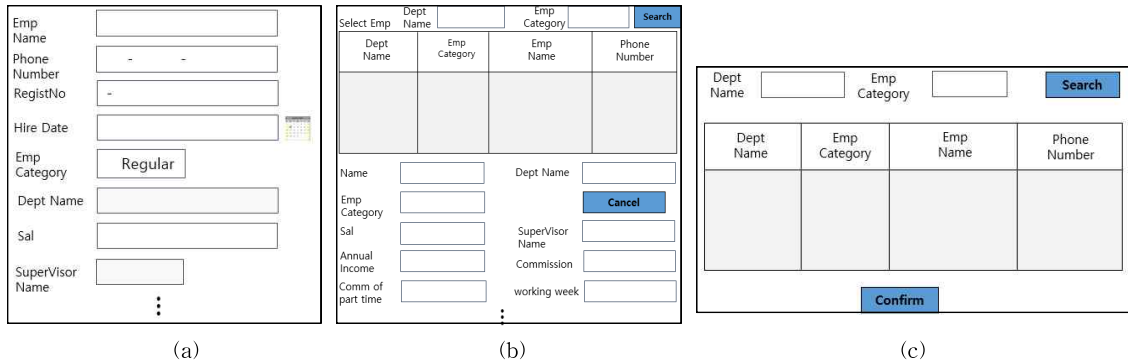


Fig. 2. Part of the Screen of Employee. (a) Screen of Register Employee, (b) Screen of Modify Employee, (c) Screen of Search Employee

Table 1. Part of Data of Employee Modify

CntCpntMapping			ContainerInf			Table								Label						
ContainerNo	ComponentDiv	ComponentNo	ContainerNo	ComponentDiv	ComponentCnt	TableNo	Name	Xsize	Ysize	Xpoint	Ypoint	Width	Height	LabelNo	Name	Text	Xpoint	Ypoint	Width	Height
106	5	19				4	Emp Category	500	200	0	1	6	1	17	Select Emp	Select Emp	0	0	1	1
106	5	20				5	Dept	150	100	0	8	3	1	18	Dept Name	Dept Name	1	0	1	1
106	5	21				6	SuperVisor	150	100	3	8	3	1	19	Emp Category	Emp Category	3	0	1	1
106	5	22												20	Emp Category	Emp Category	0	3	1	1
106	5	23												21	Name	Name	0	2	1	1
106	5	24												22	Dept Name	Dept Name	3	2	1	1
														23	Sal	Sal	0	4	1	1
TextFieldNo	Name	Length	Editable	Xpoint	Ypoint	Width	Height													
19	Sal	10	TRUE	1	4	2	1													
20	SuperVisor Name	10	TRUE	4	4	2	1													
21	Annual	10	TRUE	1	5	2	1													
22	Comm	10	TRUE	4	5	2	1													
23	CommPartTime	10	TRUE	1	6	2	1													
24	WorkingWeek	10	TRUE	4	6	2	1													

영역에서 사원관리 업무에 CRUD를 적용하여 설계된 사원의 등록, 수정, 조회 화면의 일부분을 보여준다.

Table 1은 ‘사원추가변경’ 화면 요소의 정보들 중 일부분의 데이터를 보여주며, ‘사원추가변경’ 화면은 Container 테이블에서 ‘106’으로 등록되어있다. 해당 화면에 배치되어 있는 화면 요소들의 정보는 요소별 테이블에 저장되어 있다. 화면에 소속된 화면요소들의 소속관계는 CntCpntMapping 테이블에서 확인할 수 있다. Table 1에서 CntCpntMapping 테이블의 ComponentDiv 컬럼 값 ‘5’는 TextField를 가리키며, ‘사원추가변경’ 화면에 소속된 TextField의 번호를 알 수 있다.

Fig. 3은 관리하는 화면의 데이터를 이용하여 화면 요소들을 화면에 배치하는 알고리즘을 나타낸다. 사용자가 구성할 화면을 정하면 Container 테이블에서 해당 정보를 읽어 ContainerNo를 가져온다. 이후 ContainerNo를 이용하여 ContainerInf 테이블에서 배치하고자 하는 화면에 포함된 객체별 수를 확인하

고, CntCpntMapping 테이블에서 화면의 컴포넌트 번호를 저장한다. 객체별 수에 맞추어 객체 정보를 가져오기 위해 앞에서 읽은 정보를 바탕으로 ComponentRef테이블을 참고하여 화면 요소들을 읽어서 화면에 보여준다.

Panel의 경우 Container와 동일한 순서로 PanelInf, ComponentConf, PnlCpntMapping 테이블을 활용하여 Panel에 소속된 객체들의 정보를 읽어 화면을 구성한다.

Fig. 4는 데이터 모델에 저장된 데이터를 이용하여 자바에서 구현된 실제화면의 모습을 보여준다.

#### 4. 비 교

본 절에서는 화면을 효율적으로 만들기 위해 관련 연구에서 연구되어진 방법들과 제안 방법 간에 몇 가지 항목들을 비교하였고, 그 결과는 아래의 Table 2와 같다.

개발에 소요되는 비용을 줄이기 위해서는 선행 단

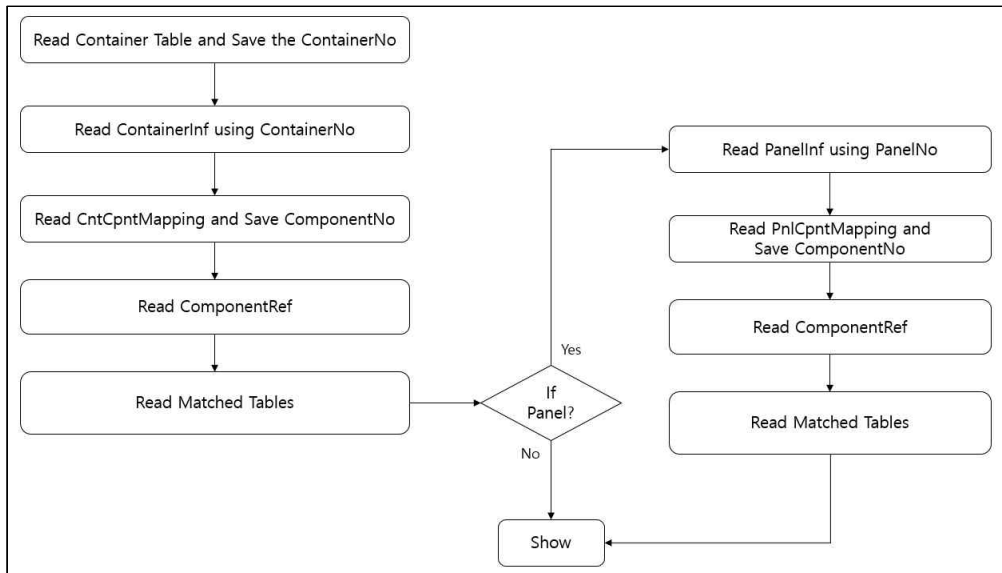


Fig. 3. Algorithm of Screen Layout.

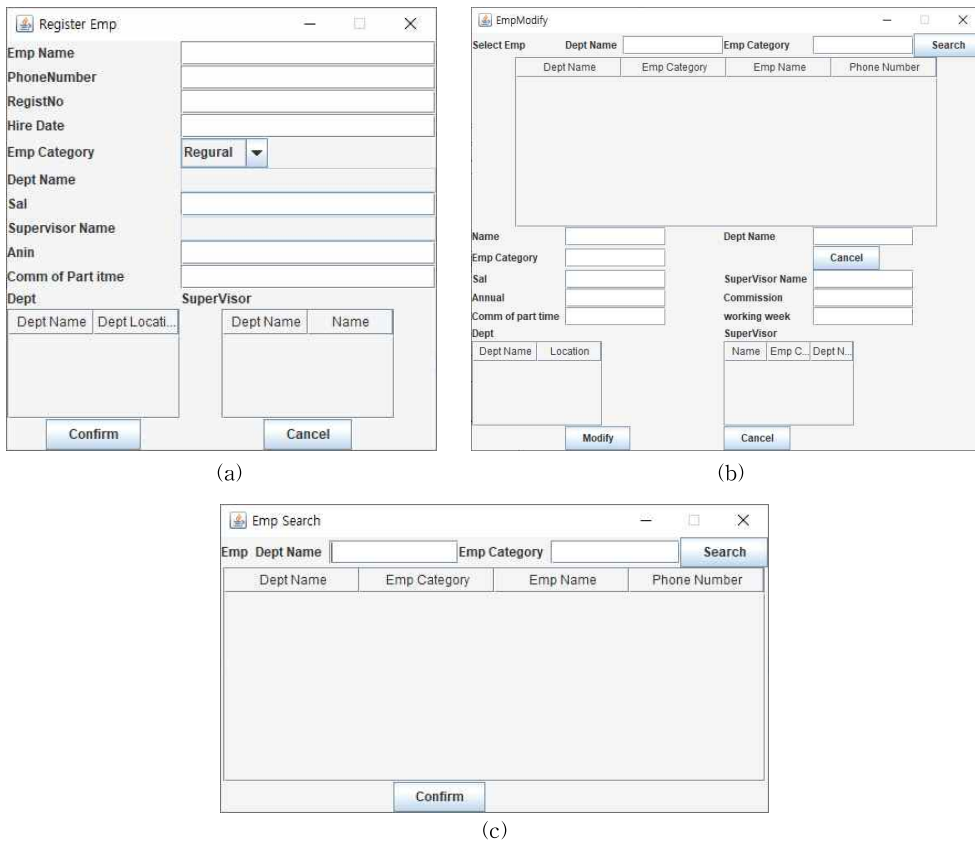


Fig. 4. Implemented Employee Screen, (a) Screen of Register Employee, (b) Screen of Modify Employee, (c) Screen of Search Employee

Table 2. Comparison Proposal Study to Related Study

	Applied Phase	whether have Visualization tool	Screen placer	whether use Database
①	Design	×	Designer	○
②	Development	○	Designer	×
③	Design	○	Designer	×
④	Development	×	Designer	×
⑤	Development	○	Developer	×

① Proposal Study, ② [16], ③ [17], ④ [4], ⑤ [5,6]  
 ○ : application, △ : partial application, × : Not Applicable

계에서 도출된 산출물을 검토하고 수정사항을 반영해야 한다. 화면을 생성하여 검토하는 단계와 검토자에 따라 수정 비용에 차이가 있을 수 있다. 개발 단계에서 화면을 생성하고 소스 코드를 생성하는 도구는 화면을 설계함과 동시에 개발이 이루어지기 때문에 수정사항이 많이 발생할 수 있다. 또한 화면 배치자가 개발자일 경우 디자인 요소로 인해 설계된 내용과의 일관성이 떨어질 수 있다.

기존의 화면생성 도구는 그래픽요소를 이용하여 시각적이고 직관적으로 화면을 생성하고 검토를 수행할 수 있다. 제안된 연구는 설계된 화면을 팔레트가 아닌 데이터로 저장하여 생성과정을 시각적으로 확인하지는 못하지만, 설계 시 정확한 수치를 지정하게 되면 설계한 화면과 동일하게 구현할 수 있다.

제안된 연구는 시스템과 별개로 화면의 내용만 저장하는 데이터베이스에 화면요소를 저장하여 사용하기 때문에 프로젝트를 수행하는 많은 사람과 쉽게 공유하고, 수정사항을 확인할 수 있다. 또한 개발 이후 화면배치의 수정이 발생할 경우에는 해당 요소의 데이터 값을 찾아가기 때문에 소스 코드를 찾아가야 하는 것 보다 쉽게 찾아 수정이 가능하다.

### 5. 결 론

본 연구에서는 ITA/EA구조에 따라 애플리케이션 영역에서 설계한 화면의 내용을 데이터로 저장하고, 데이터를 이용하여 화면을 구성하여 보여주는 알고리즘을 제안하였다. 이를 바탕으로 제안 방법과 기존의 방법들을 몇 가지 항목들에 대해 비교 수행하였고, 이를 통해 화면의 요소를 데이터로 관리하여 표현할 경우 많은 설계자간의 공유와 수정이 용이함을 알 수 있었다. 또한 생성된 화면의 소스코드는 화면

의 정보를 이용하여 구조적인 코드에 적용할 수 있어 프로그래밍 도구에 종속되지 않고 사용이 가능하였다.

이후에는 개발 단계에서 화면을 실시간으로 확인하고 수정할 수 있는 UI생성 프레임워크의 개발이 이루어져야 할 것이며, 프로그래밍 도구와 프로그램 유형에 독립적으로 적용할 수 있도록 자바 윈도우 프로그램외의 다른 유형 및 도구에서 구현하여 기능을 추가해야 할 것이다.

### REFERENCE

[ 1 ] H.K. Park, G.H. Kwon, and K.W. Lee, “A Study on a Design and Implementation of AIGS (Automatic interface code Generating System),” *Journal of The Korean Institute of Information Scientists and Engineers*, Vol. 16, No. 2, pp. 11-14, 1989.

[ 2 ] B.S. Choi, *Study on the Information Expression Techniques for Object-oriented Software Testing Tools*, Master’s Thesis of Mokwon University of Computer Engineering and Software Engineering, 2005.

[ 3 ] Y.S. Yun, J.S. Park, J.M. Jung, S.B. Eun, S. Cha, S.S. So, et al., “Automatic Mobile Screen Translation Using Object Detection Approach Based on Deep Neural Networks,” *Journal of Korea Multimedia Society*, Vol. 21, No. 11, pp. 1305- 1316, 2018.

[ 4 ] K. Adam, L. Netz, S. Varga, J. Michael, B. Rumpe, P. Heuser, et al., “Model-Based Generation of Enterprise Information Systems,”

- Proceeding of International Workshop on Enterprise Modeling and Information Systems Architectures*, Vol. 2097, pp. 75-79, 2018.
- [5] E. Schlungbaum and T. Elwert, "Automatic User Interface Generation from Declarative Models," *Journal of Travaux-Institut D Informatique*, pp. 3-18, 1996.
- [6] D.S. Lee, *Study on the Mobile Programming Automation Using UIDL(User Interface Description Language)*, Master's Thesis of Chonnam National University of Electronic Computer Engineering, 2009.
- [7] W.Y. Lee and D.S. Park, "A Design and Implementation of Parallel Programming Environment using Graphical User Interface," *Journal of Korea Multimedia Society*, Vol. 4, No. 6, pp. 579-587, 2001.
- [8] C.A. Kim, *Method of Test Automation Based on UML and Implementation*, Master's Thesis of Ewha Womans University of Computer Engineering, 2000.
- [9] A. Kraus, A. Knapp, and N. Koch, "Model-Driven Generation of Web Applications in UWE," *Proceedings of the 3rd International Workshop on Model-Driven Web Engineering*, pp. 493-496, 2007.
- [10] S. Ahmed and G. Ashraf, "Model-based User Interface Engineering with Design Patterns," *Journal of Systems and Software*, Vol. 80, No. 8, pp. 1408-1422, 2007.
- [11] H. Traetteberg, *Model-based User Interface Design*, Doctoral Thesis of Norwegian University of Science and Technology, 2002.
- [12] C.Y. Song, E.S. Cho, and C.J. Kim, "An User Interface Hierarchical Modeling Process based on Metamodel," *Journal of Korea Multimedia Society*, Vol. 11, No. 4, pp. 525-543, 2008.
- [13] J. Nichols and A. Faulring, "Automatic Interface Generation and Future User Interface Tools," *Proceeding of ACM CHI 2005 Workshop on the Future of User Interface Design Tools*, 2005.
- [14] R. Kennard and R. Steele, "Application of Software Mining to Automatic User Interface Generation," *Journal of Frontiers in Artificial Intelligence and Applications*, Vol. 182, pp. 244-256, 2008.
- [15] A.R. Puerta, "A Model-Based Interface Development Environment," *Journal of IEEE Software*, Vol. 14, No. 4, pp. 40-50, 1997.
- [16] A. Schramm, A. Preussner, M. Heinrich, and L. Vogel, "Rapid UI Development for Enterprise Applications: Combining Manual and Model-Driven Techniques," *Journal of Lecture Notes in Computer Science*, Vol. 6394, pp. 271-285, 2010.
- [17] A.M.R.d. Cruz and J.P. Faria, "A Metamodel-Based Approach for Automatic User Interface Generation," *Journal of Lecture Notes in Computer Science*, Vol. 6394, pp. 256-270, 2010.
- [18] KDPC, *Dataarchitecture Expert Guide*, Korea Database Agency, Seoul, Korea, 2013.
- [19] S.Y. Park and J.M. Yeo, "A Study on Elicitation Method of Target Businesses for Application Development," *Journal of Korea Institute of Information and Communication Engineering*, Vol. 15, No. 12, pp. 2599-2608, 2011.
- [20] S.Y. Park, T.W. Kim, and J.M. Yeo, "Study on the Business Process Procedure based on the Analysis of Requirements," *Proceeding of International Conference on Future Information Technology*, pp. 479-486, 2018.
- [21] T.W. Kim, S.Y. Park, and J.M. Yeo, "Study on the Design Process of Screen using a Prototype Method," *Proceeding of International Conference on Future Information Technology*, pp. 471-478, 2018.



김 태 우

2015년 부경대학교 컴퓨터공학과 (학사)  
2017년 부경대학교 컴퓨터공학과 (공학석사)  
2017년~현재 부경대학교 컴퓨터 공학과 박사과정

관심분야: 데이터 아키텍처, 데이터 모델링, 화면 설계



여 정 모

1980년 동아대학교 전자공학과 (학사)  
1982년 부산대학교 전자공학과 (공학석사)  
1993년 울산대학교 대학원 전자 및 전산기 공학과 (공학 박사)

1986년~현재 부경대학교 컴퓨터공학과 교수, (주)엔코 아 사외이사

관심분야: 데이터 아키텍처, ITA/EA



박 선 이

1991년 부경대학교 전자계산학과 (학사)  
1996년 부산외국어대학교 컴퓨터 공학과(공학석사)  
2012년 부경대학교 컴퓨터공학과 (공학박사)

2012년~현재 부경대학교 컴퓨터공학과 외래교수

관심분야: 데이터 아키텍처, 빅데이터, 데이터 모델링