

PCB 생산 자동화를 위한 데이터 변환 시스템 구현

Data Transformation System Implementation for the Automation of PCB Product

이승혁*, 한정수*, 김귀정**
천안대학교 정보통신학부*, 건양대학교 의공과**

Seung-Hyuk Lee(opeung@cheonan.ac.kr)*, Jung-Soo Han(jshan@cheonan.ac.kr)*,
Gui-Jung Kim(gjkim@konyang.ac.kr)**

요약

본 논문은 PCB 생산 자동화를 위한 데이터 변환 시스템을 구현한다. CAD로 설계한 데이터는 PCB 생산라인과의 데이터 호환을 이루지 못하기 때문에 데이터 호환을 갖는 자동화 시스템이 필요하다. 본 논문에서는 PCB 생산 자동화를 위한 PCB 부품 정보를 분석하고 IC 부품 정보를 DB로 구축한다. CAD 설계자의 Human error 검출 알고리즘을 개발하고, PCB 생산 라인에 적합한 데이터로 변환하는 알고리즘을 개발한다. PCB 생산라인의 부품의 정보 추가, 수정하기 위한 데이터 변환 인터페이스를 설계 및 구현한다. 기존의 수작업 공정을 자동화함으로써 처리속도를 대폭 개선하였으며, 신뢰성 있는 최적화된 데이터로 PCB 생산 자동화에 일조를 할 수 있다.

■ 중심어 : | PCB | Human Error 검출 알고리즘 | 데이터변환 알고리즘 | 데이터변환 인터페이스 |

Abstract

In this paper, we design data transformation interface for the automation of PCB product. The data designed to CAD does not exchange itself for the assembly line, so we construct an automation system which is exchangeable. To do this, we analyze the information of PCB components and construct the information of IC components as database. We also develop two kinds of algorithm; one is to detect human error and another is to exchange itself for the data which is suitable for PCB assembly line. We design data transformation interface to do addition and revision of the information for PCB assembly line. By automating existing manual processing, we are able to shorten access time and enhance reliability of the data and the efficient assembly line of PCB.

■ Keyword : | PCB | Human Error Detect Algorithm | Change Algorithm | Data Transformation Interface |

1. 서론

본 논문에서는 현재의 PCB 생산에 있어 데이터 변환을 위한 수작업을 자동화한다. 여러 가지 에러 검출을

함으로 비생산적인 현지 중소기업의 고충을 해결하고자 PCB 생산 자동화를 위한 데이터 변환 시스템을 구현한다. 기존의 개발된 데이터 변환 시스템의 경우 그래픽 설계를 강조하여 시스템을 설계하였다. 사용자 입장에

서 그래픽 설계의 이해가 없으면 사용하기 불편한 점이 많았다. 따라서 현지 중소기업은 기존의 개발된 데이터 변환 시스템을 이용하기 보다는 수작업 방식을 선호하고 있다. 현재 PCB 생산에서는 하청업체로부터 받은 PCB 설계 데이터는 CAD로 만들고 있다. 이렇게 설계된 CAD데이터는 PCB 생산 라인과의 데이터 호환이 이루어지고 있지 않는 실정이다. 즉, PCB 생산을 위해 Loader, Surface Mount Clean Machine, Screen Printer, Solder Print Tester, Chip Mounter, Tray Feeder 등의 설비 장비를 통하여 생산된다[1]. 그러나 CAD로 설계한 데이터는 이들 설비 장비와 호환성이 이루어지 않아 각 PCB에 대한 데이터를 수작업을 통하여 작업하고 있는 실정이다. 데이터 변환을 수작업에 의해 수행하다보니 Human Error가 발생하고 이로 인하여 수작업 후 그 정확도가 떨어져 불량 제품이 생산되는 경우가 빈번하여 많은 비용이 들고 있다. 또한 신뢰도가 떨어지는 경향이 있어 많은 어려움을 겪고 있다. 수작업 과정은 하나의 PCB 생산을 위한 데이터 변환만 약 8시간이 걸린다. 테스트 공정까지 포함하면 1-2일 정도가 걸린다. PCB 종류가 많을수록 그 시간과 비용은 기하급수적으로 증가한다. 현재 거의 대다수 전자 제품에 들어가는 PCB를 다량 생산할 수 있는 기능이 부족한 실정이다. 따라서 본 논문에서는 PCB 조립 공정에서 CAD로 설계된 IC 부품별 데이터를 분석을 통한 설비와의 데이터 호환을 위한 시스템을 설계, 구축하여 NC Programing 자동화로 작업능률을 극대화하기 위한 것이 그 목적이다. 이 인터페이스 구축 결과로 수작업 공정이 자동화되므로 서로 다른 Mounter(or Line 구성) 간의 NC Program Conversion(호환)이 이루어지고, BOM ↔ NC, PGM ↔ 작업지시서 비교분석으로 Human Error들을 사전에 제거할 수 있는 장점이 있을 뿐 아니라, Mounter Data & Line Balance 최적화로 생산성이 극대화될 수 있다는 결론을 얻을 수 있다. 이로 인한 PCB의 다량 생산이 가능하고, 보다 정확한 PCB 생산과 높은 신뢰도를 얻을 수 있다.

본 논문의 구성은 다음과 같다. 2장은 시스템에 대한 구조, 데이터 추출, 오류 검출, 부품 정보 추가 및 수정, 좌표 수정 및 Layer 분리와 변환 알고리즘을 기술한다.

3장은 실험 및 시스템 비교를 한다. 마지막으로 4장은 본 논문에 대한 결론과 향후 연구 방향을 제시한다.

II. 데이터 변환 자동화 시스템

1. 시스템 구조

CAD로 설계된 데이터들은 보드의 앞면 혹은 뒷면 중 어떤 위치에 결합될 것인지에 대한 데이터들을 가지고 있으며 보드에 총 부품의 개수나 그 부품의 특징들에 대한 데이터를 가지고 있다[2]. 하지만 이러한 데이터들을 PCB 생산 라인과 호환성을 갖고 있지 않다. 왜냐하면 CAD 설계자는 PCB 생산 라인을 이해하지 못한 상태에서 자신만의 설계 방법에 따라 설계하기 때문이다. 이는 똑같은 PCB를 설계할 때에도 설계자에 따라 다른 결론의 CAD 데이터가 나타날 수 있다는 의미가 되므로 이러한 상의한 데이터를 PCB 라인에 적합한 데이터로 변환하는 과정이 필요하다. 따라서 PCB 제조업체는 이러한 정보를 다시 수작업으로 하나하나 모두 변환하여 PCB 라인에 적합한 데이터로 작성해야하는 부담을 안고 있다. 이러한 비효율적인 작업은 데이터를 변환하는 PCB 생산업체의 고비용과 생산성 감소에 직접적인 연관을 가지고 있다. 뿐만 아니라 수작업을 통해 발생할 수 있는 Human Error는 엄청난 손실을 초래할 수 있다.

[그림 1]에서 CAD 데이터의 정보 분석을 통해 부품 정보를 추출한다. PCB 생산 라인의 전 과정을 분석하여 필요한 PCB 부품에 대한 정보를 획득하며 부품별로 데이터를 분류한다[3]. 이렇게 CAD에서 분석된 부품 정보와 PCB 라인에서 분석된 부품 정보가 정확히 일치한가를 확인한다. 만약 일치하지 않을 경우 이를 정확한 부품 정보를 갖도록 수정하게 된다. 이렇게 수정된 데이터들은 최소한 PCB 부품 정보를 가지게 되며 데이터 저장소에 저장되게 된다. 저장소에 구축될 정보 안에는 품목 명, 품목번호, 위치, 각도, 규격, 수량 등의 데이터들이 포함된다. 저장소는 인터페이스를 통한 부품 검색을 위해 키워드를 정의하고 데이터 변환 프로그램을 위한 준비를 완료한다. 이 과정에서 CAD 데이터의 오류

수정을 하게 되는데 CAD 설계자에 의해 설계된 데이터가 설계자의 Human Error에 의해 잘못된 데이터를 가질 수 있기 때문이다. 이런 문제를 해결하기 위해 Human Error 검출 알고리즘을 이용하여 저장 데이터를 검증 받는다. 검증 과정에서 Error가 발생하면 이를 인터페이스 사용자에게 알린다. CAD 데이터 중에서 Error가 발생한 데이터를 다시 입력받게 된다. 이런 과정을 통해 Error가 없는 데이터에 대하여 필요한 부품의 정보를 추가하게 된다. 이 부품 정보는 이미 PCB 생산 공정을 분석하여 얻은 데이터로부터 가지고 오게 된다. 또한 CAD 설계자는 PCB 보드의 좌표를 자신의 방법에 따라 정하는 반면, PCB 생산 라인은 보드의 한쪽 모서리를 원점으로 갖고 좌표를 계산한다. 따라서 CAD 데이터의 좌표 값들을 PCB 생산 라인에 적합한 좌표 값들로 변환하게 된다. 좌표 값 변환은 인터페이스를 통해 사용자에게 현재의 CAD 데이터의 좌표 값이 어디를 원점으로 정하여 설계되었는지를 알려준다. 사용자는 PCB 생산 라인의 원점을 확인하는 방법으로 원점을 수정하게 된다.

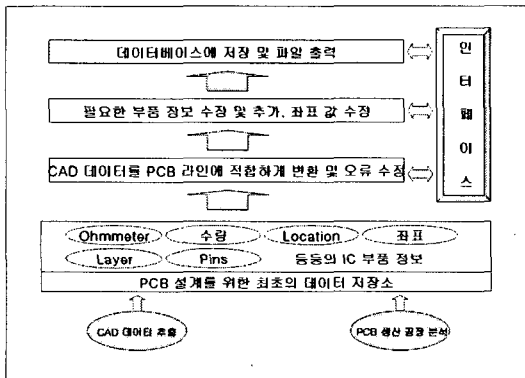


그림 1. 데이터베이스 변환 시스템 구조

2. CAD 와 PCB 생산 라인에서 데이터 추출

CAD 데이터는 CAD 설계자의 편의에 의해 PCB 생산 라인과는 관계없는 불필요한 데이터를 가지고 있다. 이런 불필요한 데이터는 PCB 생산 라인의 공정 분석을 통해 얻을 수 있다. 분석된 결과에 따라 CAD 데이터 중 실제적으로 필요한 데이터만을 추출하게 된다. 뿐만

아니라 CAD 데이터 중에서 설계자의 편의에 의해 삭제된 일부 데이터들이 PCB 생산 라인에서는 유용한 정보일 수 있다. 따라서 이렇게 추가, 삭제된 데이터에서 불필요한 데이터는 삭제하고 필요한 데이터는 다시 추가하여 추출하게 된다[4]. 예를 들어 CAD 설계자는 Vendor에 대한 정보를 상세히 기록하여 CAD 데이터에 삽입할 수 있다. 하지만 Vendor는 PCB 생산에 어떤 영향도 미치지 않기 때문에 불필요한 Vendor에 대한 정보는 삭제하게 된다. 또한 'Specification'란 데이터가 없을 경우 이는 앞에서 이미 CAD 설계자가 'Specification'에 대한 데이터를 정의하였기 때문이다. 하지만 데이터 저장소에 그대로 저장하게 된다면 이는 NULL로 인식하게 된다. 따라서 앞에 이미 정의된 데이터를 찾아내어 삽입하게 된다. 이처럼 예를 들어 설명하였지만 이러한 필요 혹은 불필요한 데이터는 수없이 많이 있다. 이를 하나하나 찾아내어 수정하여 PCB 데이터 저장소에 저장하게 된다. [그림 2]는 CAD 데이터를 추출하여 데이터 저장소에 저장하여 CAD 설계자에 의한 CAD 데이터 오류나 Human Error를 제거하는 단계를 보여주고 있다. 이 단계에서는 저장소에 오류 없는 데이터를 저장하기 위함이다.

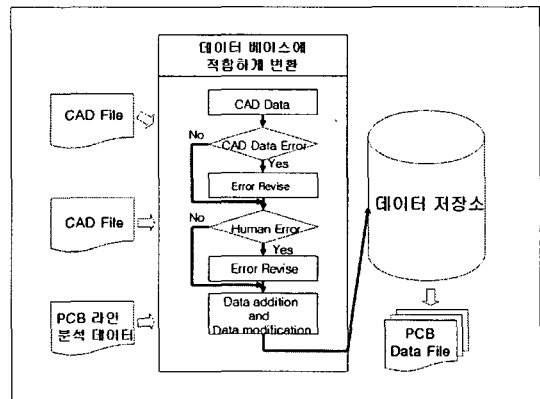


그림 2. 저장소에 적합 데이터로 변환

3. PCB 데이터 저장소

PCB 부품저장소에는 부품 위치, 각도(X,Y값), 규격, 품목 명, 품목 번호, 구성 수량 등의 정보로 구성된다. 이

정보들은 CAD 데이터 중에 변환 과정에 필요한 정보만을 자동으로 추출하게 된다. 저장소에는 두개의 Table을 가지고 있다. 이는 CAD 설계자가 두개의 파일 형태로 CAD 데이터를 만들기 때문이다. 각각의 Table은 차후 각종 Error 검출 알고리즘과 데이터 추가 및 수정을 통해 하나의 Table로 결합되게 된다[5]. 이렇게 결합된 Table은 또다시 Layer에 값에 따라 두개의 파일로 출력 된다. 이 두 파일은 각각 서로 다른 PCB 생산 라인에서 적용되게 된다. Layer는 PCB 보드에서 각 부품이 보드의 앞면 혹은 뒷면에 장착될 것인지를 알려주기 때문이다. 이들 정보를 바탕으로 PCB 설비 장비가 인식할 수 있는 데이터들로 변환하게 되는 데이터의 구조는 [표 1]과 같다. [표 1]의 Location No와 RefDes를 각각 기본키로 하며 같은 위치정보를 갖는다. 이 기본키를 통해 하나의 Table로 결합하게 된다. 이는 PCB 생산라인에서 하나의 텍스트 파일로 입력을 요구하기 때문이다. 또한 인터페이스를 통하여 변환된 데이터들이 시뮬레이션을 통하여 원래의 목적에 맞도록 잘 되었는지를 테스트할 수 있는 검사 프로그램을 삽입하여 설비 장비에 입력되기 전에 테스트한다.

표 1. PCB 부품 저장소의 데이터 구조

Table	Field name	Data Type	설명
1	Specification	String	저항
	Q'ty	long	수량
	Location No	String	위치
	Approved Vendor 1 st	String	
	Approved VendorParts No 01	String	
	Approved Vendor2 nd	String	
2	Approved VendorParts No 02	String	
	PartType	String	부품
	RefDes	String	위치
	PartDecal	String	부품
	Pins	long	핀수
	Layer	bool	앞, 뒤
	Orient	long	각도
	X	long	좌표
	Y	long	좌표
	SMD	bool	부품
Glued	bool	부품	

4. 데이터 오류 검출 및 정렬

CAD 데이터를 PCB 데이터로의 변환을 위해 우선 CAD 데이터 자체의 오류 검출 및 수정해야한다. CAD

데이터는 CAD 설계자에 의해 만들어졌기 때문에 언제든지 Human Error를 포함할 가능성이 있기 때문이다. 이런 오류 검출을 위해 몇 가지 검증 알고리즘을 개발하였다. 이 검증 알고리즘은 CAD 데이터 내의 Error 유형을 파악하고 이를 수정하기 위한 방법들을 마련하였다. 검증 알고리즘은 CAD 데이터에서 품목 명, 품목 번호, 위치, 각도, 규격, 수량 등이 PCB 생산 설비에 의해 이미 저장되어 있는 품목 명, 품목번호, 위치, 각도, 규격, 수량 등의 정보와 정확히 일치한지를 파악한다. 이것이 일치하지 않을 때는 Human Error로 간주하고 사용자에게 CAD 데이터가 잘못된 것이라는 메시지를 보내고 새로운 정보를 입력받도록 한다. 또한 CAD 데이터 내의 품목 명과 품목 번호가 일치하지 않을 때, 수량과 PCB 보드가 수용할 수 있는 수량이 초과하였을 때, 위치 정보의 개수와 부품 수량이 다를 때 등의 여러 가지 잘못된 데이터들을 검출하여 새로운 값들을 입력받는다. 입력된 정보는 부품 저장소에서 기존의 정보와 치환하게 된다. 하지만 이렇게 입력된 값들 또한 잘못된 데이터일 가능성이 있기 때문에 다시 데이터 오류 검증 알고리즘을 이용하여 정확한 데이터인지 검증받아 이 데이터들이 오류를 포함하지 않으면 다음 단계인 정렬 알고리즘을 사용하여 데이터를 변환하게 된다. [그림 3]은 데이터 오류 검증 알고리즘을 보여주고 있다.

```

Input CAD data in Database
Database field are item_name, item_number,
location_NO, coordinates, amount
save each filed of location_NO data
For( All Q'ty )
  if( # of Q'ty same Location_Number)
    make records for # of Q'ty in database
    separate Location_NO in each number
Input standard radian
FOR(all items)
  IF(item_number != amount)
    output ("Human Error")
    input new item_number
  ENDF
    
```

그림 3. 데이터 오류 검증 알고리즘

정렬 알고리즘에서는 DB에 데이터들을 저장한 다음 각 item별로 분류한다. Location_Number(위치 수)의 경우 한 Database의 field 안에 여러 개의 위치 정보를 포함하고 있다. 따라서 이를 각각의 위치 정보를 분류하여 새로운 field안에 저장하여 Q'ty(부품 수)와 Location_Number(위치 수)가 일치하는지 확인한다. 만약 일치 하지 않다면 새로운 Q'ty(부품 수)와 Location_Number(위치 수)를 입력받는다. [그림 4]는 Database에 저장된 Location_Number 데이터들을 정렬하여 새로운 Database로 변환된 모습을 보여주고 있다. 마지막으로 item의 개수와 총 부품의 개수를 비교한다.

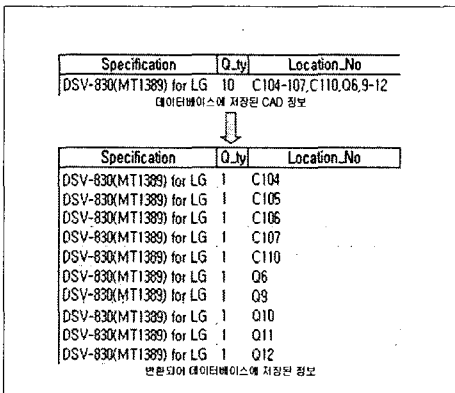


그림 4. 정렬되어 Database에 저장된 데이터

5. 부품 정보 수정 및 추가

PCB 생산 라인의 전 과정을 분석한 PCB 부품에 대한 데이터와 CAD 데이터를 비교하여 부품 정보를 수정 및 추가한다. 이를 위해 우선 Database의 Table1에 저장되어 있는 Location No와 Table2에 저장되어 있는 RefDes를 서로 비교하여 같은 데이터를 찾는다. Location No와 RefDes의 값은 PCB 보드에 어떤 위치에 IC부품을 조립할 것인지에 대한 정보를 가지고 있다. 만약 두 Table을 비교하여 같은 데이터가 있다면 Table1의 Specification, Q'ty와 Table2의 Orient, X, Y, Layer를 Database 내의 다른 Table에 저장하게 된다. 물론 Location No와 RefDes의 값도 같이 저장하게 된다. 이는 최종 PCB 생산라인에 입력해야 할 데이터

가 Location No와 X, Y, Orient, Specification, Layer를 포함하고 있기 때문이다. 즉, 이 값들은 실질적인 PCB 보드에 어떤 위치에 어떤 부품이 들어갈 것인가를 알려주는 최소한의 정보인 것이다[6].

CAD 데이터의 부품 각도인 Orient의 경우 CAD 설계자에 의해 임의의 각도 값이 주어질 가능성이 있다. 왜냐하면 CAD 설계자는 PCB 생산 라인의 부품 각도를 분석하지 않고 단순히 부품 표준 각도에 의해 CAD로 작성하였기 때문이다. 따라서 본 논문에서 미리 PCB 생산 라인의 전 과정을 분석한 데이터를 가지고 서로 비교하여 이들의 각도에 대한 표준 각도와 PCB 생산라인에서 원하는 각도를 분석한다. 이 값들이 일치하지 않을 때 이를 수정하여 변환하게 된다. 이렇게 하나의 Table에 저장된 데이터들은 좌표 수정과 Layer 분리를 위한 단계의 준비를 마치게 된다.

6. 좌표 수정과 Layer 분리

하나의 Table로 저장된 데이터에서 PCB 부품별 데이터는 CAD와 같은 데이터들을 포함하고 있지만 CAD 데이터의 좌표 값(x,y)과 PCB 생산 라인에서의 좌표 값(x,y)는 서로 다른 체계로 되어있기 때문에 각각의 좌표 값이 어떤 위치를 중심으로 설계되었는지 확인할 필요가 있다. CAD 설계자는 좌표 값을 PCB 보드의 중간정도의 위치를 원점으로 정한다[7]. 반면에 PCB 생산라인에서는 PCB 보드의 모서리 부분을 원점으로 사용하고 있다. 따라서 CAD 설계자가 어떤 위치를 원점으로 정했는지 원점 알고리즘을 통해 PCB 보드의 네 군데의 모서리 중에서 가장 가까운 모서리를 찾아 그 거리를 계산한 후 인터페이스를 통해 사용자에게 거리에 대한 오차를 입력받도록 요구하고 이 오차에 대하여 원점 알고리즘은 모든 좌표 값을 수정하게 된다. [그림 5]는 CAD설계자의 임의의 원점을 계산하여 이동하는 모습을 보여주고 있다.

좌표 값이 수정된 데이터에서 Layer 값에 따라 새로운 다른 Table에 저장하게 된다. Layer의 값은 PCB 보드의 앞면 혹은 뒷면에 IC부품을 조립할 것인지에 대한 값을 가지고 있기 때문이다. Layer의 값이 Top의 경우 앞면에 조립되고 Bottom의 경우 뒷면에 조립되기 때문

에 각각을 분리하여 저장하게 된다. 누락이 된 데이터에 대하여 Layer의 값을 입력받도록 사용자에게 알려 새로운 Layer의 값을 입력받는다. 이는 PCB 생산라인이 Top과 Bottom으로 구분되어 있기 때문이다. 또한 생산라인은 하나의 텍스트 파일로 데이터를 입력하게 되는데 그 또한 Layer 값에 따라 하나 파일로 입력받도록 되어있다. [그림 6]은 데이터 변환 알고리즘을 보여주고 있다. 이 알고리즘에서는 우선 Table1과 Table2를 하나의 테이블로 조인 연산한다. 하나의 테이블을 좌표 수정과 Layer 분리를 통하여 데이터 변환을 하게 된다.

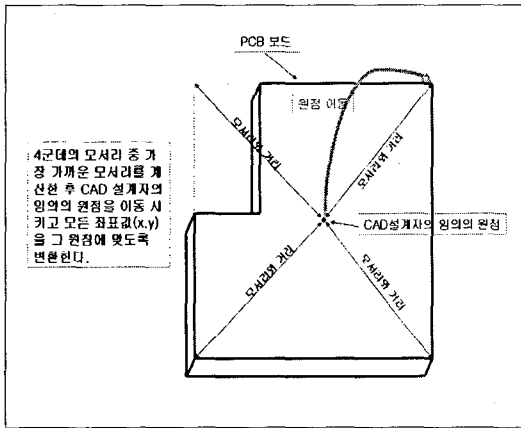


그림 5. 원점 수정

```

Location No save in Database Table1
RefDes, X, Y, Orient save in Database Table2
FOR(Last Recode)
  IF (Location No of Table 1 =RefDes of Table 2)
    Table 1 JOIN Table2
  END IF
  Input standard (X,Y)
  conversion (X,Y)
  IF (CAD_data(radian) != Real Radian)
    input new standard Radian
    Do radian transformation
  ENDIF
  IF (Layer = top)
    save of Top_Table
  ELSE
    save of Bottom_Table
  ENDIF
  
```

그림 6. 데이터 변환 알고리즘

III. 실험 및 성능 평가

1. 데이터베이스 및 파일 출력

지금까지는 여러 개의 데이터베이스 Table에 저장된 정보들을 통합하고 [그림 7]은 출력된 정보들을 보여준다[8]. 데이터베이스는 사용자의 편의성을 고려하여 흔히 사용하는 Access로 설정하였다. 이는 사용자가 데이터 변환에 있어 그 과정을 쉽게 알아볼 수 있으며 Excel 파일로의 변환도 자유로워 각각의 부품에 대한 정보를 사용자가 쉽게 찾아 볼 수 있기 때문이다. 최종 PCB 생산 라인에 필요한 부품 데이터만을 읽어 최종 텍스트 파일로 출력한다. 텍스트 파일 출력이 있어 한 라인에 하나의 부품 정보만을 출력한다. 부품 정보의 출력 데이터는 Location No, Specification, Orient, X, Y, Layer 순이며 각각의 값은 Tab 간격으로 구분하였다.

파라미터	위치(라디안)	색상(라디안)	도움말(라디안)
0206	R1C1170-10CV, SOT-223, SMD, 101C	100.0	102.862500
0206	R1C1170-10CV, SOT-223, SMD, 101C	100.0	102.862500
0201	CF, 20Pin, 0.5mm, SMD, RA, 5410A-30MOLEX	193.7	16.261200
0202	TOPIn, 1mm, RA, SMD, 21F, BOTTOM, MOLEX	0.0	82.855000
0202	CF, 8Pin, 1mm, SMD, RA, 65100-65150 CABLE	270.0	4.787800
0202	CV, 8Pin, 1.25mm, RA, SMD, 53700-04MOLEX	0.0	106.482000
0202	CV, 8Pin, 1.25mm, RA, SMD, 53700-04MOLEX	270.0	109.482000
0202	CF, 20Pin, 0.5mm, SMD, RA, 52689-20MOLEX	90.0	88.956000
0204	CV, 8Pin, 1.25mm, RA, SMD, 53700-04MOLEX	270.0	109.482000
C13	1600, COG, 2PF, C, 50U	100.0	28.376000
C14	1600, COG, 2PF, C, 50U	100.0	28.440000
C15	1600, COG, 2PF, C, 50U	100.0	28.376000
C16	1600, COG, 2PF, C, 50U	100.0	28.440000
C17	1600, COG, 2PF, C, 50U	100.0	28.376000
C18	1600, COG, 2PF, C, 50U	100.0	28.440000
C22	1600, COG, 5PF, C, 50U	90.0	76.764000
C212	1600, COG, 5PF, C, 50U	100.0	28.440000
CA	1600, COG, 20PF, J, 50U	100.0	28.376000
C200	1600, COG, 20PF, J, 50U	270.0	28.827500
C210	1600, COG, 20PF, J, 50U	90.0	16.582500
C201	1600, COG, 33PF, J, 50U	100.0	91.750000
C202	1600, COG, 33PF, J, 50U	0.0	104.869000
C1	1600, COG, 33PF, J, 50U	100.0	28.376000
C04	1600, COG, 50PF, J, 50U	90.0	50.296000
C05	1600, COG, 50PF, J, 50U	90.0	58.792000
C12	1600, X7R, 1000P, K, 50U	100.0	28.376000
C08	1600, X7R, 1500P, K, 50U	270.0	62.822000
C7	1600, X7R, 2200P, K, 50U	100.0	28.376000
C203	1600, X7R, 3300P, K, 50U	100.0	91.750000

그림 7. 최종 출력 결과(Top Layer)

2. 인터페이스

인터페이스 설계에 있어 데이터베이스에 저장된 데이터들을 사용자가 쉽게 접근하여 데이터의 검증이나 확인을 받을 필요가 있다. 이는 CAD 설계자의 잘못된 설계에 의한 Human Error를 방지할 수 있을 뿐만 아니라 수정이 필요한 데이터에 대해서도 수정을 쉽게 할 수 있도록 하기 위함이다. 이러한 인터페이스가 없다면 잘못된 PCB 데이터에 의해 되돌릴 수 없는 막대한 손실을 얻을 수 있다. 따라서 본 논문에서는 사용자에게

쉽게 검증 및 확인 할 수 있고 시스템에 의해 변환된 과정을 쉽게 알아볼 수 있도록 인터페이스를 설계하였다. 이 인터페이스는 네트워크나 인터넷이 특별히 필요 없는 관계로 PCB 제작자의 편의성을 고려하여 운영체제는 윈도우를 이용하였으며, 데이터베이스에 접근할 수 있는 인터페이스의 개발을 위해 데이터베이스는 Access를 이용하여 OLE-DB로 데이터베이스 응용 프로그램을 만들어 접근하였다. 여기서 Access를 이용한 것은 사용자가 쉽게 사용할 수 있다는 장점도 포함된 결과이다. OLE-DB는 COM 서버의 기능을 이용하여 COM 클라이언트 프로그램을 만들 것이므로 MFC보다는 ATL을 이용하는 것이 효율적이지만 ATL이 사용자 인터페이스에 관련된 기능을 거의 지원하지 않는 관계로 MFC와 ATL(Active Template Library)을 모두 이용하였다[9]. [그림 8]은 본 논문에서 구축한 데이터 변환 시스템의 결과이며, [표 2]는 각 단계별의 수행 내용을 보여주고 있다.

그림 8. 데이터 변환 결과

표 2. 단계별 내용들

단계별	실행 내용
1단계	테이블정렬, 오류수정
2단계	부품정보수정 및 추가
3단계	CAD 설계자의 원점 확인
4단계	원점 및 X, Y 좌표 값 수정
5단계	데이터베이스 기록
6단계	내용 확인
7단계	파일 출력

3. 기존 시스템과 비교

본 논문에서 제시한 시스템과 기존의 데이터변환 SMT 시스템[10]을 비교한다. 비교 기준은 오백 개의 이상 부품 데이터를 갖는 CAD 데이터를 PCB 생산 라인에 적합한 데이터로 변화하는 과정으로 한다. 변환 과정은 총 35회 실시한다. 비교 기준은 사용자가 그들의 작업을 수행하는데 시스템이 얼마나 도움을 주는가의 정도와 사용하는데 얼마나 쉽고 편안가에 따라 소프트웨어 시스템의 질을 평가한다[11]. 평가 목록으로 효율성의 증가, 생산력의 개선, 에러의 감소, 훈련의 감소, 사용자 수용의 증가로 한다[12]. 효율성의 증가는 시스템이 좋은 인간 공학 설계를 가지고 있고, 사용자가 수행하는 방법이 시스템의 인터페이스와 일치한지를 평가한다. 생산력의 개선은 사용자를 혼란시키지 않고 수행하는 작업의 집중도를 평가한다. 에러의 감소는 잘못된 사용자 인터페이스 때문인데, 사용자 에러를 줄일 수 있는 것처럼 모순, 애매함 등을 피할 수 있지는 정도를 평가한다. 훈련의 감소는 인터페이스를 얼마나 배우기 쉬운지 정도를 평가한다. 사용자 수용의 증가는 사용자가 기존의 시스템과 논문에서 제안한 시스템 중 어떤 시스템을 선호하는 정도를 평가한다. 이 평가는 현지 PCB 생산 업체에게 기존의 시스템과 본 논문에서 제시한 시스템을 비교 평가하는 항목을 제시한다. 제시한 항목은 만족도를 100으로 하고 총 8명의 사용자에게 질의하였다. [그림 9]는 기존 시스템과 본 논문이 제시한 시스템을 비교한 그래프이다.

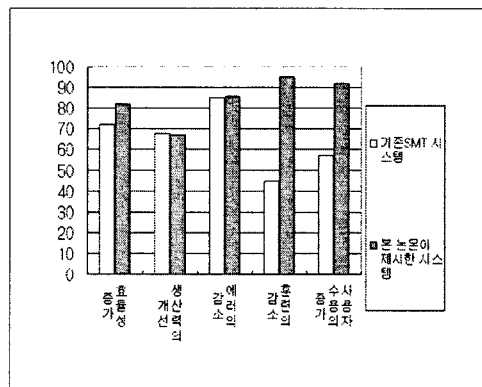


그림 9. 기존 시스템과 비교

IV. 결론

본 논문의 목적은 사용자가 쉽고 편리하게 CAD 데이터를 PCB 생산 라인에 적합한 데이터로 변환하는 것이다. 그동안 이런 과정은 수작업에 의해 1-2일이 소요되었으며 직접 눈으로 테스트하는 과정을 거쳐야 했기 때문에 많은 시간과 불량 생산할 가능성을 갖고 있었다. 하지만 본 논문에서 이러한 과정을 획기적으로 보완하였다. 정확한 PCB 생산을 위해 여러 가지 검증 알고리즘과 신뢰성을 확보할 수 있도록 PCB 생산 전 과정을 분석하였다. 또한 CAD 데이터 내에 Human Error를 검출하고 PCB 생산 전 과정을 분석한 데이터를 비교하여 최소한의 오류까지 제거하였다. 이렇게 오류를 제거한 데이터들을 변환 알고리즘을 통해 CAD 데이터를 PCB 생산라인에 사용하는 데이터로 변환할 수 있도록 하였고, 이를 사용자가 쉽게 접근하고 변환 결과물을 확인할 수 있는 인터페이스를 설계 및 구현하였다. 기존의 그래픽 설계 위주의 데이터변환 시스템을 획기적으로 개선하여 사용자가 인터페이스를 통해 쉽게 접근할 수 있도록 시스템을 구현하였다. 본 논문에서 제시한 시스템은 사용자가 클릭 몇 번으로 PCB 라인에 적합한 데이터를 얻을 수 있었다. 이로써 정확하고 효율적인 PCB 생산을 위한 이기종간의 데이터 호환을 통한 공장 자동화를 완성하였다. 사용자의 편의성과 생산라인에 필요한 데이터만을 적합하게 출력하는 인터페이스를 구축하여 기존의 시스템보다 작업 능력을 향상시킬 수 있었다.

향후 연구는 데이터 변환이 이루어진 후 그 결과에 대한 검증을 인터페이스를 통해 사용자가 직접 확인하였으나 이를 자동으로 검증하는 알고리즘 개발이 요구 된다.

참고 문헌

[1] <http://www.qmaxtest.com>
 [2] 한건희, PCB 설계 패키지 프로그램 CAD star, 세운출판사, 1999.
 [3] 송영재, 객체지향모델링과 CBD중심 소프트웨어

공학, 이한출판사, 2004.

[4] C. Batini, s. Ceri, and S. B. Navathe, Conceptual Database Design : An Entity-Relationship Approach, Benjamin Cummings, 1992.
 [5] Y. Kambayashi, W. Kim, D. S. Reiner, and D.S. Batory ed, Processing Cyclic Queries, In : Query Processing in Database Systems., Springer-Verlag, 1985.
 [6] D. Heoy and C. E. Leiserson, "A layout for the shffle-exchange network." Proc. of the 13th Annual ACM Sym. on the Theory of Computing, pp.335-342, 1981.
 [7] 김경수, PCB 회로 설계, 홍릉과학출판사, 2004.
 [8] K. E Emam and N. H. Madhavji, A Field Study of Requirements Engineering Practives in Information Systems Development, York, England, 1995.
 [9] 박광우, Visual C++.Net Programming Bible, 삼양미디어, 2004.
 [10] <http://www.smtkorea.co.kr>
 [11] M. Lea, Evaluation User Interface Design, user Interface Design for Computer System, Halstead Press, Wiley, 1988.
 [12] T. Mandel, The Elements of User Interface Design, Wiley, 1997.

저자 소개

이 승 혁(Seung-Hyuk Lee)

준회원



• 2004년 2월 : 천안대학교 컴퓨터학과(공학사)
 • 2004년 3월~현재 : 천안대학교 정보기술대학원 석사과정

<관심분야> : 소프트웨어공학, OOP, CBD

한 정 수(Jung-Soo Han)

종신회원



- 1990년 : 경희대학교 전자계산공학과(공학사)
- 1992년 : 경희대학교 전자계산공학과(공학석사)
- 2000년 : 경희대학교 전자계산공학과(공학박사)

• 2001년~현재 : 천안대학교 정보통신학부 조교수
 <관심분야> : 소프트웨어공학, CBD, S/W형상관리

김 귀 정(Gui-Jung Kim)

정회원



- 1994년 : 한남대학교 전자계산공학과(공학사)
- 1996년 : 한남대학교 전자계산공학과(공학석사)
- 2003년 : 경희대학교 전자계산공학과(공학박사)

• 2001년~현재 : 건양대학교 의공학과 조교수
 <관심분야> : CBD, CASE 도구, 컴포넌트 검색