



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

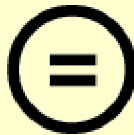
다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공 학 석 사 학 위 논 문

공공기관 연구시설장비 활용성 향상을 위한

웹기반 일정계획 시스템설계 :

객체 모델링 기법의 적용

(Designing a Web-based Scheduling System for Increasing Public
Research Facility Equipment Effectiveness: Application of
Object-Modeling Technique)

2016년 12월

부 경 대 학 교 대 학 원

기 술 경 영 협 동 과 정

이 채 윤

공 학 석 사 학 위 논 문

공공기관 연구시설장비 활용성 향상을 위한

웹기반 일정계획 시스템설계 :

객체 모델링 기법의 적용

(Designing a Web-based Scheduling System for Increasing Public
Research Facility Equipment Effectiveness: Application of
Object-Modeling Technique)

지도교수 신 승 준

이 논문을 공학석사 학위논문으로 제출함.

2016년 12월

부 경 대 학 교 대 학 원

기 술 경 영 협 동 과 정

이 채 윤

이채윤의 공학석사 학위논문을 인준함

2017년 2월 24일



위 원 장 공학박사 옥 영 석 (인)

위 원 공학박사 최 승 욱 (인)

위 원 공학박사 신 승 준 (인)

목 차

I. 서론	1
1.1 연구의 배경 및 필요성	1
1.2 연구의 목표	3
1.3 연구의 내용 및 방법	5
II. 본 론	7
2.1 전통적 데이터 모델 고찰	7
2.1.1 레코드리반이 데이터 모델	8
2.1.2 의미 데이터 모델	13
2.1.3 객체지향 방법론과 데이터 모델링	15
2.2 객체지향 방법론과 데이터 모델링	23
2.2.1 객체지향방법론	23
2.2.2 복합개체	29
2.2.3 객체지향데이터 모델의 개발 의의	30
2.3 객체지향데이터모델의 개발 절차	35
2.3.1 OMT기법	35
2.3.2 객체 모델링 절차	37
2.3.3 클래스 위주의 평가 기준	39

2.4 시험·분석·연구 장비현황 및 공동 활용 관한 선행연구.....	40
2.4.1 정부 투자 시험·분석·연구 장비 현황.....	40
2.4.2 시험·분석·연구 장비 공동 활용에 관한 선행 연구 및 사례.....	43
2.4.3 시험·분석·연구 장비의 공동 활용 파급효과 분석.....	44

III. 일정계획 개념적 설계 47

3.1 일정시스템.....	47
3.1.1 생산기능 및 형태의 기본개념.....	48
3.1.2 생산계획의 형태.....	48
3.1.3 일정계획.....	49
3.1.4 일정계획 기능.....	53
3.2 자재소요계획(MRP)과 도면 발행.....	55
3.2.1 MRP의 목표와 특색.....	55
3.2.2 MRP의 입출력.....	56
3.2.3 도면 발행 계획.....	57
3.3 객체지향 데이터 모델의 설계.....	59
3.3.1 문제설명서 작성과 클래스의 식별.....	59
3.3.2 데이터 사전과 클래스 관계도.....	62
3.3.3 클래스의 속성 추가.....	63
3.3.4 객체지향 데이터 모델의 도출.....	68

VI. 대형설비 생산 일정계획 시스템 설계 70

4.1 데이터 모델 설계의 개요 70

4.1.1 대형 설비 생산의 특징 70

4.1.2 적용기업의 분석 72

4.1.3 데이터 모델 설계의 방향 75

4.2 객체 모델링과 데이터 모델의 도출 78

4.2.1 대형 설비 생산의 특징 78

4.2.2 데이터사전과 클래스 관계도 80

4.2.3 클래스 속성의 확인 81

4.2.4 일반화를 통한 클래스의 단순화 87

4.2.5 데이터 모델 도출 89

4.3 적용기업 데이터 모델에 대한 논의 91

4.3.1 적용기업 데이터 모델의 특징 91

4.3.2 적용기업 데이터 모델의 시사점 94

V. 활용시나리오

5.1 활용시나리오 설정 97

5.2 운영 시나리오 98

5.3 기대효과 104

VI. 결론 및 향후 연구	106
6.1 논문의 요약	106
6.2 논문의 공헌	111
6.3 한계점 및 극복방안	112
참고문헌	114



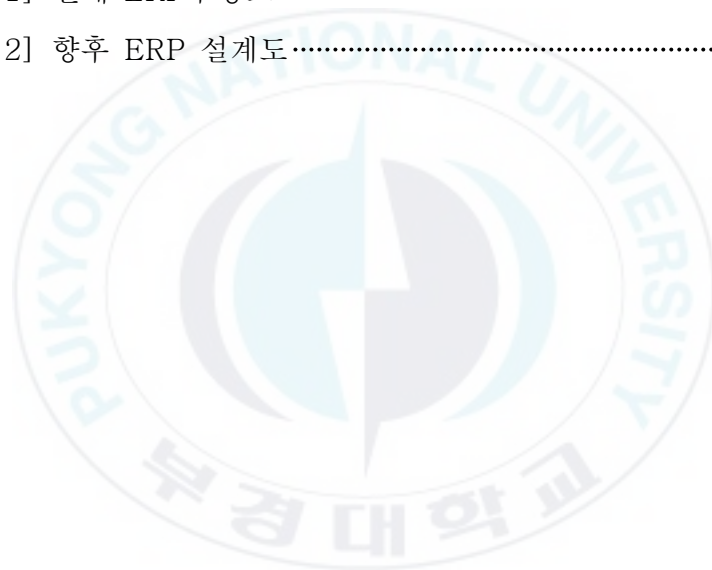
표 목 차

[표 2-1] 도서대출 시스템의 관계형 모델	20
[표 2-2] 도서대출 시스템의 종속 관계	20
[표 2-3] 도서대출 시스템의 관계형 모델 의미 표현	20
[표 2-4] 객체지향 관련 용어	24
[표 2-5] 연구시설장비의 활용목적별 분류	41
[표 2-6] 시험분석연구장비의 공동 활용에 관한 선행연구	43
[표 2-7] 분석 대상 장비 현황	44
[표 2-8] 대상장비를 공동 활용할 경우 가동률 증가에 의한 절감 효과 ..	45
[표 3-1] 생산 일정계획의 문제 설명서	61
[표 3-2] 일정계획 클래스의 데이터 사전	62
[표 4-1] 문제 설명서	79
[표 4-2] 확인된 객체클래스	80
[표 4-3] 데이터 사전	81
[표 5-1] ERP 설비정보화면	100
[표 5-2] ERP 장치 및 생산단위, 자재군 등록화면	101
[표 5-3] ERP 공정단위 및 도면(군)	102
[표 5-4] ERP 자재 및 대시보드	103

그 림 목 차

[그림 2- 1] 실행시의 포괄적 테이블 접근.....	12
[그림 2- 2] 도서관 대출 시스템의 개념적 모델.....	18
[그림 2- 3] 캡슐화와 메시지 전달.....	24
[그림 2- 4] 시스템개발과정.....	26
[그림 2- 5] OODB와 RDB 통합방식.....	34
[그림 2- 6] OMT 표기법.....	36
[그림 2- 7] 지역별 시험분석연구장비의 분포 상황.....	42
[그림 3- 1] 일반적인 생산계획 활동.....	49
[그림 3- 2] 일반적인 일정계획 구성도.....	52
[그림 3- 3] MRP의 체계.....	56
[그림 3- 4] MRP의 입출력.....	57
[그림 3- 5] 제품과 생산계획 데이터를 포함하는 CAD DB.....	58
[그림 3- 6] 객체 클래스 식별 순서.....	59
[그림 3- 7] 생산일정계획의 클래스 관계도.....	63
[그림 3- 8] 대일정계획에 관한 정보흐름도.....	64
[그림 3- 9] 소일정계획에 관한 정보흐름도.....	65
[그림 3-10] 속성을 포함한 객체 클래스 관계도.....	67
[그림 3-11] 객체지향데이터모델 도출.....	69
[그림 4- 1] Level 초기정보흐름도.....	73
[그림 4- 2] 설비의 클래스 계층도.....	75
[그림 4- 3] 기능적 분해를 통한 데이터 모델링.....	76
[그림 4- 4] 객체지향 설계의 활용.....	77

[그림 4- 5] Leve2 데이터흐름도	78
[그림 4- 6] 속성을 포함한 객체 클래스 관계도	82
[그림 4- 7] 설비 클래스의 속성	83
[그림 4- 8] 속성을 포함한 객체클래스 관계도	88
[그림 4- 9] 일반화 관계의 적용	89
[그림 4-10] 일반화 관계를 포함한 객체 클래스 관계도	91
[그림 4-11] 도출된 데이터 모델	92
[그림 5- 1] 현재 ERP구성도	97
[그림 5- 2] 향후 ERP 설계도	104



Designing a Web-based Scheduling System for Increasing Public Research Facility Equipment Effectiveness : Application of Object-Modeling Technique

Lee- Chae Yun

기술 인프라의 체계적 관리는 국가 및 기업의 경쟁우위 창출 원동력이 되는 현시대에 있어 전략적으로 매우 중요한 의미를 갖는다. 이러한 맥락에서 기술 인프라 요소 중 하나인 공공 시험·분석·연구 시설장비(이하, 시설장비)는 국가 차원에서 활용성 제고를 위한 활용 방안을 비중있게 검토할 필요가 있고, 시설장비의 체계적 구축과 관리 효율화를 위한 노력이 요구된다. 그러나, 시설장비를 제공하는 공급기관과 이를 사용하는 수요기업간 의사소통 및 정보교환이 수작업 및 오프라인 방식에 의존하고 있는 것이 현실이다. 이로 인하여 사용시점의 예측 불가 및 유희시간 증가 등 문제를 야기하여 시설장비의 효율적 활용을 저해하고 있다.

이를 극복하기 위하여, 본 논문에서는 시설장비 활용성 향상을 위한 실시간 협업 및 정보공유를 가능하게 하는 웹기반 통합 일정계획 시스템을 설계한다. 본 시스템은 일정관리, 장비관리, 주문관리 및 고

객관리 업무를 온라인상에서 일괄적으로 수행하고, 생성 및 교환되는 정보를 통합 데이터베이스에서 저장 및 처리하는 기능을 갖춘다. 본 시스템을 위하여 객체지향 모델링 기법을 활용하여 표준화된 시스템 모델을 설계하고, 업무프로세스 개선 모델과 시스템 구축 모델을 설계한다. 본 논문의 결과물은 기존 전사적자원관리 시스템과 통합되어 시설장비에 대한 수요기업의 사용가시성 및 공급기관의 정보가시성 및 수요예측성 향상을 가능하게 하여 시설장비의 활용성 향상에 이바지 할 것이다.



Abstract

Management of technology program, Pukyong National University

The systematic management of technology infrastructure has been increasingly recognized to be a critical strategic meaning as it currently represents the driving power for creating competitive advantage of nations and enterprises. In this sense, public test, analysis and research facility equipment (hereafter, facility equipment), which is one major asset of technology infrastructure, should be effectively dealt with, systematically constructed and efficiently managed to increase its overall equipment effectiveness. However, the manual and off-line communication was the unique way to communicate between the suppliers who provide the equipment in the public sector and the users who utilize the equipment in the industrial sector for reserving and using facility equipment. This communication way caused the invisibility of available usage time and the increase of idle time of the equipment, thereby decreasing the overall equipment effectiveness.

To overcome this problem, this thesis presents the design of a web-based scheduling system that enables real-time collaboration and information sharing for increasing the facility equipment effectiveness. The designed system performs scheduling, facility management, order management and customer-relationship management in an on-line platform, and carries out the processing and storing of relevant information in an integrated database. For this

purpose, this thesis includes the design of standardized system models, business process models and system implementation models by using an object-modeling technique. The deliverable of the thesis will contribute to increasing the facility equipment effectiveness by enhancing the visibility of users and the transparency and predictability of suppliers by means of the integration with a legacy Enterprise Resource Planning system.

Keywords : Research Facility Equipment System, object-oriented data model, databases, equipments, technology infrastructure,



I. 서론

1.1 연구의 배경 및 필요성

국가 연구 개발업이 증가함에 따라서 연구시설장비의 구축도 함께 증가하여 공동 활용 및 중복투자 방지 등 연구시설장비의 범정부적인 효율적 운영의 필요성이 제기되어 왔다. 한정된 예산으로 연구개발 성과를 높이기 위해서는 연구시설장비에 대한 체계적 구축과 관리 효율화를 위한 노력을 통해 장비활용 효율성을 높일 수 있는 일정계획 관리 시스템 필요하다는 지적이 있어 왔다. 각 공공기관 대상의 기업들을 효율적으로 관리 및 운영하기 위한 경영 기법은 미흡한 것이 현실이다. 우리나라의 연구개발사업의 효율성이 선진국에 비하여 떨어지는 것은 연구시설장비 관리제도 결여에서 비롯된 측면이 크다.

본 논문에서는 최종적으로는 과학기술분야의 연구개발 사업으로 구축되는 국가연구시설장비의 효율적 관리를 위하여 연구시설장비의 체계적 구축과 공동 활용을 촉진하기 위하여, 최종제품이 납기내에 생산될 수 있도록 여러 구성품들과 부품을 생산하고, 그때 소요되는 자재와 도면에 대한 조달 일정을 수립 가능한 설계로, 사용자 객체지향 시스템 (OMT : Object Modeling Technique)을 기반한 설계로 가동률 최대화, 재고수준 최소화, 고객서비스 최대화를 목적으로 하는 모델링구현으로 작업부하균형(품질)과 런타임 오버헤드(납기) 해결 가능한 실제세계의 객체를 직접 모델과 1:1 대응시킴으로써 실제와 모델간에 존재하는 의미론적 차이를 최소화 할 수 있고, 최근 소프트웨어 개발 및 유지보수에 가장 큰 문제인 재사용성을 증대 시킬 수 있도록 표준화된 객체지향데이터모델

설계 도출하도록 한다.

특히, 이러한 모듈들은 프로세스 사이의 독립적인 객체의 흐름을 중점적으로 다루는 거래지향(transactional-oriented)의 모델링과 실제 시스템의 구성요소의 행위와 기능에 기초한 객체지향(object-oriented)의 모델링이 모두 가능하다. 객체지향적 방법은 실세계의 객체를 직접 모델과 1:1 대응시킴으로써 실제와 모델간에 존재하는 의미론적 의미를 전달할 수 있고, 최근 소프트웨어 개발 및 유지보수에 가장 큰 문제인 재사용성을 증대시킬 수 있다.

따라서, 객체지향적 소프트웨어 기법 중에서 모델기반, 컴포넌트기반, 최종사용자에 의한 해법조합 등의 방법론이 이용되고 있으며, 아직까지 데이터 모델에 대한 표준화된 모델이 모호한 상황에서 표준화된 모델을 정의하고 설계하고 서비스 표준화된 모델을 개발하는데 있어 가장 효과적으로 활용될 수 있는 객체지향모델링 절차를 제안한다. 모델링 단계는 OMT(Object modeling Technique)을 이용하고, 향후에 더 향상된 서비스의 표준화된 모델을 개발하는데도 활용될 수 있을 것으로 예상된다.

1.2 연구의 목표

최근 WTO/TBT, 한·미 FTA 타결 등 자유무역 체제가 급속히 진행되고 있으며, 이에 따라 과학기술 R&D의 경쟁력이 국부 창출의 결정적 요인으로 작용할 것으로 평가되고 있다. 또한 대내·외 산업 환경 변화로 강제 시험·인증분야가 확대되고, 무역 및 산업에서 그 영향력이 크게 증가하고 있는 실정으로 시험·분석서비스 분야에서도 시장 개방요구가 거세질 것으로 예상되고 있다. 이로 인해 국내 기업의 시험·분석·연구 장비 활용에 대한 수요가 증가하는 추세이다. 일례로 자국의 산업 보호를 위해 수출국의 관련 인증취득을 강화하고 있는 유럽, 미국, 중국의 경우 CE, UL, CCC 등을 통해 관련 표준 등을 더욱 세분화 하고 있으며 RoHS, WEEE, REACH 등 국제 환경규제 확대는 물질 분석의 정밀성을 더욱 더 높여가고 있다. 정부는 이러한 인식하에 시험·분석·연구 장비의 첨단화와 유허장비의 활용성 제고를 위한 정책을 추진하고 있는데 첨단 장비의 확보 여부가 기술개발의 성패를 좌우하는 경향이 심화되어, 국내 기관 간에 동일한 장비를 중복 구매하는 경우가 빈발하고 있기 때문이다. 이러한 이유로 국내 기관간의 장비 공동 활용은 매우 중요한 문제임과 동시에 활용성 제고를 위해 필요한 사안이라고 할 수 있다. 모든 의미를 인정하고도 우리나라의 경우에는 이러한 관리 시스템 구축의 필요성을 느끼고 있지만 아직까지 체계적인 시스템을 구축하고 있지는 못한 실정이다.

정보기술의 발달과 더불어 정보의 홍수 속에 수많은 기업은 비용을 절감하고 이윤을 극대화하기 위한 방법으로 정보시스템을 앞다퉈 도입하기에 이르렀고, 초창기 정보시스템은 기존의 업무 - 영업, 생산, 구매, 자재, 인사, 급여, 회계 등-를 제한된 환경에서 단위 업무 중심으로 사용하였고 이를 개발하기 위한 소프트웨어 또한 프로그래밍 중심적인 구조적 설계 방법

에 의존하여 개발함으로서 많은 시간과 비용이 발생 하였다. 결국 개발 생산성이 저하되고 소프트웨어 품질 또한 저하되는 문제가 발생하였다.

업무의 처리과정에서 그 업무의 범위와 규모가 커지고 고객의 다양한 요구 사항에 따른 시스템의 복잡성을 처리할 필요성을 느끼게 되었다. 특히 물리적인 시스템의 분산, 동시성(Concurrency), 반복성(Replication), 보안, 결점보완, 시스템들의 부하에 대한 균등화(Load balance)와 같은 반복해서 발생하는 구조적 문제 대한 처리가 필요하게 되었다.

기업체에서 제품생산이나 서비스 창출을 위해 일정계획을 수립함에 있어서 오늘날의 발달된 데이터베이스 기술들이 일정계획의 질을 높이는데 크게 기여할 수 있음은 두 말 할 나위가 없다. 증가하고 있는 연구시설장비 수요에 보다 효율적으로 대처하고 양질의 데이터를 얻기 위해서는 장비들에 대한 체계적이고 전문적인 유지관리 시스템 구축이 필요하며 본 논문에서는 웹기반 통합 일정시스템 설계 필요성을 살펴보고 연구시설장비 활용 방안도 검토해보도록 한다. 하나의 대안으로 기존 데이터베이스를 이용하여 제품생산이나 서비스 창출을 위해 기업의 생산능력을 특정 과업이나 활동에 배정하는 일정계획을 통하여 구체적인 일정계획을 수립하여 장비의 가동률을 최대화하고, 재고수준 최소화하며, 고객서비스는 최대화 할 수 있는 일련의 처리과정을 가진 응용 시스템이 개발되어야한다.

현재에 와서는 기업들의 다양한 요구가 발생하고 이에 능동적으로 대처하기 위한 개발 방법론이 필요하였다. 많은 회사에서 소프트웨어에 대한 전략적인 가치가 증가됨에 따라 소프트웨어 생산의 자동화, 소프트웨어의 시간과 비용을 절감, 소프트웨어의 질을 향상시킬 수 있는 기술을 모색하고 있다.

따라서 지속적으로 한계점이 지적되어 온 관계형 데이터 모델을 보완하기 위해 시스템분석 및 설계기법으로 각광받고 있는 객체지향 접근법을 통해

생산시스템의 일반적인 데이터 모델을 도출하고, 그것을 대형설비를 생산하는 업체의 구체적인 경우에 적용시키는데 목적이 있다.

이러한 복잡하고 다양한 요구사항을 대체할 수 있는 개발 방법론이 객체지향방법론이므로 객체지향 데이터 모델을 도출하는 것은 그 과정에서 객체지향 분석설계가 이루어지는 만큼 데이터 모델링 이외에도 시스템의 추상화, 모듈성, 모듈 내 응집력 그리고 정보은닉과 같은 부서적 효과를 수반한다는 점에서 객체지향 접근법의 활용은 의미를 더 하게 된다.

대형설비를 주문 받아 생산하는 보다 구체적인 생산시스템의 일정계획을 위한 데이터 모델은 일반적인 경우보다 데이터의 동태적 속성을 더 많이 포함하게 될 생산시스템이 특성상 객체지향방법론을 통해 정태적속성과 동태적속성을 동시에 데이터 모델링을 설계해보고자 한다.

1.3 연구의 내용 및 방법

본 논문은 현재 클라이언트/서버 환경에서 사용하고 있는 고객사를 대상으로 고객사에서 제공하는 수주 정보의 연계 및 수주 정보의 부정확성이나 문제점을 분석하고, 장·단기 장비가동계획, 기간별 수주 정보, 수주 대비 출고관리, 판매계획 대비 매출실적 등의 업무 영역에 대한 자세한 수준의 프로세스 분석을 통한 개선점을 찾아내고 현행 ERP 시스템의 관리 모듈에서 수행 가능한 업무의 통합과 재사용이 가능한 정보의 제공 가능성에 대해 분석해 보고자 한다. 또한 분석된 AS-IS를 기반으로 TO-BE 프로세스 모델 설계 등의 과정을 거쳐 객체지향 방법론 설계 도구인 객체지향방법을 이용해 이론적 고찰과 선행연구에 관하여 논의, 생산시스템의 일정 계획에 관계되는 클래스들과 그들간의 관계를 모형화하고, 객체지향데이터모델링을 통해 각 클래스의 속성과 오퍼레이션을 정의, 연구시설장비(대형설비)

를 대상으로 데이터모델을 평가하여 가능한 업무의 통합과 일관성 있는 정보제공 가능성 분석 모델링을 설계하고자 한다.

2 장에서는 객체지향데이터모델에 대해 살펴보고자 한다. 관계형 데이터모델과 그 한계점 그리고 표현력 제고를 통한 의미적 측면의 수용 요구를 살펴보고, 그 이후에 등장한 의미데이터 모델 및 객체지향 데이터모델에 대해서 차례로 고찰한 후, 객체지향 시스템과 객체지향 방법론, 그리고 객체 모델링 절차에 대해서 문헌 연구를 실시하고 시험·분석·연구 장비 현황 및 공동 활용에 관한 선행 연구도 살펴본다.

제3장에서는 생산일정계획 문헌을 기반으로 전반적인 의미를 살펴보고 생산 일정계획을 위한 데이터베이스 객체지향 데이터 모델링 과정을 통해 설계한다. 객체지향 설계 과정을 통해 일반적인 생산시스템의 일정계획에 사용되는 객체 집단과 그들간의 관계를 모형화하기 위해 본 연구에서는 객체 모델의 도출을 위해 객체 모델링 기법(Object Modeling Technique : OMT)을 사용하게 된다.

제4장에서는 대형설비를 생산하는 기업을 대상으로 앞장에서 설계된 데이터 모델을 적용해보고 이를 확정하여 일정계획용 데이터베이스를 설계한다. 대형설비 생산일정계획은 설비 발주자의 요구사항이나 하위사업부의 구성에 따라 기본적으로 단속적 생산시스템의 일정계획과 유사한 과정을 갖고 있으며, 그 생산 활동이 규모가 크고 장기간에 걸쳐 수행되므로 프로젝트 관리의 성격도 포함하게 된다. 그러한 성격을 고려하여 대형 설비를 주문받아 생산 납품하는 업체의 일정계획용 데이터베이스를 위한 객체지향 데이터 모델을 도출한다.

제5장에서는 대형설비를 생산 이용하는 부산테크노파크의 활용 및 운영 시나리오를 서명하고 이를 사용하는 기업의 기대효과를 알아본다.

끝으로 대형설비 일정계획 데이터베이스를 위한 객체지향 데이터 도출 과

정의 요약 과 연구의 한계점 및 앞으로의 연구방향도 결론과 함께 제6장에 제시하게 된다.

II. 본론

2.1 전통적 데이터 모델 고찰

최대로 데이터베이스의 구조는 논리적 구조와 물리적 구조로 나누어 고찰될 수 있다. 논리적 구조는 사용자 입장에서 데이터의 본질이나 형태를 정의하는 측면이며, 물리적 구조는 디스크 상에서 데이터가 저장되는 구조 및 저장된 구조를 표현한 것을 데이터 모델이라 하는데, 데이터모델을 사람이나 조직에 있어서 의미를 가지는 현실세계의 일부분을 모형화하기 위해 사용되는 지적 도구로 정의하고 있다.

모형화 도구로서 데이터 모델은 현실세계에 대해 일관성을 가지도록 하는 제약조건들을 구체화하는 일련의 구조체를 제공하며, 표현의 도구로서 데이터 모델은 데이터 설계자와 사용자가 분석자 또는 시스템 개발자간의 의사소통을 담당하게 된다.

전통적으로 데이터베이스 스키마 설계를 위한 데이터 모델은 데이터의 정태적인 속성들을 모형화 하는데 사용되었다. 그러나 전통적인 데이터 모델링에 정보시스템 설계를 위한 소프트웨어 공학의 관점인 기능적 분석을 결합하려는 시도가 계속되었고, 그러한 개념을 지원하는 데이터 모델 기법과 이론들이 많이 연구되었다. 데이터 모델을 레코드기반의 데이터 모델, 의미데이터모델, 그리고 객체지향데이터모델로 나누어 그 변천을 논의하고자 한다.

2.1.1 레코드 기반의 데이터 모델

계층형, 네트워크, 관계형 데이터 모델을 포함하는 이른바 전통적 데이터 모델은 사용되는 데이터 구조가 컴퓨터상에서의 물리적 데이터 표현에 비교적 근사하고, 궁극적으로 포인트 필드값을 가진 레코드의 집합으로 데이터를 보고 있다는 견지에서 레코드기반으로 볼 수 있다. 이들 전통적 데이터 모델에서는 데이터 모형화의 대상이 되는 실체가 레코드나 튜플로 표현되었을 때, 별개의 것으로 존재하는 단순한 실체와 실체에 연관된 속성이거나 여러 실체의 집합체인 실체간에 전혀 구별이 없다는 점을 객체기반의 데이터 모델과 구별되는 특성으로 지적하고 있다.

2.1.1.1 계층형 및 네트워크 데이터 모델 (Hierarchical Data Model & Network Data Model)

데이터베이스의 논리적 구조가 트리(Tree) 형태로 표현되며 E-R 모델의 엔티티를 레코드 타입(Record Type)으로 표현한다. 또 E-R 모델의 1:N 관계를 두 레코드 타입간의 부모-자식(Parent-Child) 관계로 표현한다.

계층형 데이터베이스에서 속성의 집합을 세그먼트(Segment)라고 한다. 계층 구조는 부모 세그먼트(Parent Segment)라고 하는 데이터 세그먼트에서 시작한다. 부모 요소 밑에는 여러 개의 자식 세그먼트(Child Segment)가 있다. 계층형 데이터 모델은 루트 세그먼트(Root Segment)에서 출발하여 다음 레벨로 점차적으로 내려오면서 가장 왼쪽의 세그먼트를 우선적으로 방문한다. 따라서 가장 빈번하게 접근되는 세그먼트를 트리의 왼쪽에 가깝게 배치함으로써 데이터에 접근하는 시간을 최소화할 수 있다.

계층형 모델은 1:N의 관계를 가지는 대용량의 데이터베이스 처리에 매우 강력한 방법으로, 주어진 기능 내에서 찾고자 하는 데이터 항목을 빠르게

찾을 수 있는 장점이 있고 단점으로는 사용자가 데이터가 저장된 물리적 구조에 대한 지식을 가지고 있어야 하고, 접근의 유연성이 부족하다. 데이터간의 모든 관계는 데이터베이스가 처음에 설계될 때 정해져야만 한다. 또한 계층형 모델은 1:N의 기준에 맞지 않는 일반적인 관계를 구현하기가 매우 어렵다.

데이터 구조가 복잡해지면서 계층형 데이터 모델의 단순한 구조는 오히려 단점이 되었다. 따라서 네트워크형 데이터 모델은 데이터를 표현하는 데 있어 제한적인 트리 구조가 아닌 그래프 형태를 기반으로 한다. 이는 현실 세계의 데이터 구조를 표현하기 위한 필요성에 의해 만들어진 것이다.

즉, 네트워크 데이터 모델은 계층형 모델이 확장된 형태로 간주할 수 있다. 그 결과 나타나는 레코드 구조는 트리가 아닌 네트워크가 되며, 레코드 종류를 연결하는 링크 종류에 고유한 이름과 방향표시가 붙여지게 된다. 네트워크 모델은 계층형과 달리 기존의 두 레코드 종류가 공통된 자식 레코드 종류를 가짐으로써 둘 사이의 다대다 관계가 구축 될 수 있다. 그러나 연결되어 있는 링크의 연결 흐름이 복잡하다는 문제를 가진다.

이들 두 모델에서 레코드를 연결하는 링크는 물리적 차원에서의 포인터를 거의 그대로 반영한 것에 지나지 않으므로 사용자가 보는 것은 데이터의 논리적 구조가 아니라 물리적 저장구조 그 자체이다. 그러므로 이는 논리적 구조와 물리적 구조를 분리하는 역할을 하는 데이터모델 이라기 보다는 다양한 파일을 조직화하기 위한 일종의 파일관리 모델이라고 보는 견해도 있다. 실제로 데이터 모델이라하면 다음절에 소개될 관계형 데이터 모델을 일컫는 것이 일반적이다.

2.1.2 관계형 레코드 기반의 데이터 모델 (Relational Data Model)

관계형 데이터 모델은 1970년 코드(E.F.Codd)에 의해 소개된 모델로, 단순하고 균일한 데이터 구조를 가지고 있어 많이 사용되고 있는 모델이다. 관계형 데이터 모델은 표현하고자 하는 엔티티의 모든 데이터와 데이터 사이의 관계를 2차원의 테이블(Table) 형태로 기술한다.

테이블의 각 행(Row)은 하나의 엔티티를 나타내고, 테이블의 각 열(Column)은 엔티티의 각 속성을 나타낸다. 하나의 테이블은 보통 여러 개의 행들로 구성되므로 엔티티 집합은 테이블구조로 표현된다고 할 수 있다. 모형화의 단순성으로 인해 폭넓게 전파되어 사용되고 있으며, 오늘날의 상용데이터베이스 시스템의 대부분이 이를 기반으로 하고 있다. 또한 선택, 프로젝트, 카티션 곱, 합집합, 그리고 차집합등과 같은 연산들을 포함하는 관계형 대수는 테이블 전체를 연산자의 매개변수로 간주함으로써 질의 처리에 있어 강력한 이점을 제공한다.

이러한 처리의 처리는 일차 술어 연산의 형태인 관계형 연산을 기반으로 하는 비절차 언어인 SQL을 통해 데이터 처리 분야에 있어 사실상의 표준으로 자리잡고 있다.

관계형 데이터 모델에 대한 대표적인 제약조건으로는 개체 무결성 제약과 참조무결성 제약의 두 가지를 들 수 있다. 개체 무결성 제약이란 동일한 실체에 대응하는 하나의 릴레이션을 생각할 때 그 실체를 지칭하는 튜플은 단 하나만 존재해야 한다는 것이다. 즉 이를 통해 값의 유일성을 보장받게 된다. 참조 무결성 제약은 외래키 개념에서 비롯된다. 한 릴레이션의 주키를 다른 릴레이션에 전입하여 한 속성을 공통적으로 보유함으로써 두 릴레이션이 서로 연결된다. 이 때 외래키(foreign key)의 값은 해당 주키가 취하고 있는 값들 중 하나와 일치하거나 널(null)값이어야 한다는 것이 참조 무결성 제약으로, 튜플들 사이에서 값의 일관성을 유지하기 위해서 필

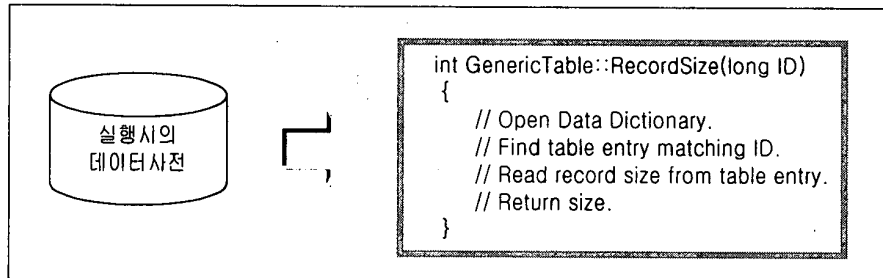
요하다. 이러한 데이터 구조와 제약조건에 대한 요구들은 정규화(normalization)라는 과정을 통해서 성취된다.

이 외에도 관계형 데이터 모델이 가지는 장단점은 다음과 같다.

- ① 관계형 대수와 간결한 의미 표현은 시스템 내의 질의 최적화를 가능하게 한다. (예:DDL과 DML이 간단하여 비전문요원도 쉽게 사용)
- ② 데이터베이스의 물리적 요소로부터 논리적 요소를 분리시킴으로써 일반 사용자도 데이터베이스를 이해하고 다룰 수 있게 한다.
- ③ 데이터들 간의 복잡한 관계를 개념적으로 분명하고 간단하게 표현하며, 강력한 데이터 조작능력을 제공한다.
- ④ 데이터의 첨가, 삭제, 수정이 쉽고, 미래의 정보요구에 신축성있게 대응할 수 있다.
- ⑤ 관계형 데이터 모델의 단점은 기억용량이 많이 필요하고 그 구조 자체의 평면성(flatness)에 기인하다. 이는 데이터간의 관계와 연결에 포함된 중요한 의미적 정보들이 소실될 수 있다는 점에서 전체적인 표현력의 부재로 볼수 있고 정보추출에 있어서 상당한 시간이 소요되는 점이다

2.1.1.3 객체 모델로서의 관계형 데이터 모델

[1] Campbell과 Joseph은 관계형 데이터 모델의 각 테이블은 사실상 그 테이블 고유의 동작(오퍼레이션)을 내부적으로 가지고 있는 것으로 보았는데, 이는 관계형 데이터 모델이 일종의 객체모델이라는 것을 의미한다. 예를들면 한 테이블에 대해서 그 테이블을 선택하거나, 수정하거나, 또는 테이블 피라미터를 리턴하기 위한 메소드가 포함된 것으로 볼 수 있다는 것이다. 그러한 오퍼레이션들은 모든 테이블을 동일한 단일체의 데이터 스토어로 취급함으로써 포괄적인(non-object-specific) 형태로 구현되어 있는 것이다.



[그림2-1] 실행시의 포괄적 테이블 접근

출처: J. A. Campbell et al., "The Object-Oriented Design and Implementation of a Relational Database Management System", Journal of Object Oriented Programming, Jul.-Aug., pp.44.

[그림2-1]은 관계형 데이터베이스가 실행(run-time)되어 사용자가 특정 테이블에 접근했을 때의 상황을 나타낸 것이다. 그림에 나타난 범용 목적의 RecordSize 메소드는 테이블에 대한 식별자를 받아서, 그 식별자에서 비롯되는 테이블 정보를 찾기 위해 실행 데이터 사전을 검색하고, 발견된 레코드 사이즈를 돌려주는 오퍼레이션을 수행하게 된다. 그러나 그 동일한 오퍼레이션은 개별적인 형태로 취급될 수도 있다. 그렇게 함으로써 각 테이블에 국한된 세부적인 사전 지식을 가지는 이점을 취할 수 있다. 이것은 객체지향 개념에 포함되어 있는 개별화된 지식과 동작과 가운 것이다. 따라서 각 테이블은 자신의 데이터 구조는 물론 데이터 조작 오퍼레이션에도 포함하게 된다.

2.1.2 의미데이터 모델

관계형 데이터 모델이 많이 활용되고 있으나, 많은 데이터 모델 설계자들에 의해 테이블의 형태로 바로 이식될 수 없는 문제들을 개념적으로 모형화함에 있어서는 충분치 못하다는 것이 지적되어 왔다. 데이터베이스 내에 보다 많은 의미적 측면들을 함축시키고자 하는 노력으로 연구된 여러 데이터모델들을 의미 데이터 모델이라고 분류할 수 있다. 데이터베이스 모델링의 수준을 외부적(external) 혹은 논리적(logical)수준, 개념적(conceptual)수준, 그리고 내부적(internal) 혹은 물리적(physical)수준의 수준으로 나누어 생각할 때, 관계형 데이터 모델의 테이블은 논리적 수준의 객체들간의 모든 관계를 표현해 주기에는 부족함이 있다 하더라도 물리적 수준의 모델로 쉽게 이식될 수 있는 편리한 수단이 될 수는 없다. 반면 의미데이터 모델의 범주에 속하는 모형화 기법들은 모델에 표현된 데이터 구조를 물리적 구조로 이식시킬수 있는 수단을 제공할 뿐 아니라 모형화의 대상(enterprise)을 바라보는 사용자의 논리적 시각에 대응하는 객체와 관계를 구체화하는 수단이 된다는 점에서 개념적이고 외부적인 수준으로 볼 수 있다. 의미 데이터 모델에서 표현하고자 하는 의미적 측면들은 데이터 추상화의 개념에서 출발하는데, 일반화 집단화, 분류화 그리고 연관 등을 포함한다.

2.1.2.1 일반화(Generation)

일반화개념은 유사한 객체들 중에 존재하는 몇몇 유사한 점들을 객체들간의 유사점을 강조하는 상위의 종류를 형성하여 표현한다. 일반화 과정을 통해서 나타나는 의미는 하나의 상위 클래스가 여러 개의 하위 클래스와 연관되는 클래스들의 계층으로 나타난다. 따라서 그 계층을 따라 상위 수

준으로 올라감에 따라 일반화되고, 하위 수준으로 내려감에 따라 구체화되는 형태로 데이터가 추상화된다. 또한 일반화 계층을 통해서 속성의 계층이 상위 클래스로부터 하위클래스로 이루어지게 된다.

2.1.2.2 집단화(Aggregation)

집단화 개념은 하위 수준 타입간의 관계가 상위 수준 타입으로 간주될 수 있는 의미적 관계를 나타낸다. 이는 하나의 집단 객체를 여러 개의 구성 (componet) 객체로부터 생성하게Te는 추상화의 개념을 지원하는 것으로, 집단화 관계를 가지는 상위 객체에 대해 하위 객체는 IS_PART_OF의 관계를 가지는 것으로 표현된다. 이 개념은 다시 여러 속성들을 집단화하여 모음으로써 별개의 객체를 생성하는데도 사용될 수 있다.

2.1.2.3 분류화(Clsssification)

분류화는 여러 객체의 집합이 상위 수준의 객체 클래스로 간주되는 형식의 추상화이다. 객체 클래스에 대해 각 객체들은 IS_INSTANCE_OF의 관계로 표현된다. 구체화 개념이 이미 존재하는 타입의 명세(specification)로부터 또 하나의 타입 정의를 도출하는 반면, 분류화 개념은 하나의 특정 객체의 타입을 명시해 주는 메커니즘을 제공한다.

2.1.2.4 연관(Association)

연관은 멤버 객체들간의 관계과 상위 수준의 집합 객체로 간주되는 추상화의 형태이다. IS_MEMBER_OF의 관계가 연관 개념을 실현시키게 되는데 연관을 표현하는 술어(predicate)를 만족하는 멤버들을 전체집합인 상위객체에서 뽑아내는 의미를 나타낸다. 따라서 집단화의 경우 연관의 특수한 형태 태압의 속성들을 명세화하는 수단이라면, 연관은 특정한 타입에

속하는 객체의 집합을 값으로 갖는 또 하나의 타입을 정의하기 위한 메커니즘이 된다는 점에서 구별이 가능하다. 이와 같은 의미 정보들을 데이터 모델이 내포함으로써 전반적인 표현력제고, 무결성 유지, 모형화의 효율성 증대 등의 이점을 얻을 수 있기 때문에 의미 데이터 모델에 대한 연구가 활발히 이루어졌다. 그러나 실제 데이용되었다.

터베이스 시스템으로 구현되는 예는 그리 많지 않았으며, 상용 데이터베이스로 자리 잡지는 못하였고, 주로 관계형 데이터베이스 스키마를 설계하기 위한 지표로서 사용되었다.

2.1.3 객체지향 데이터 모델

2.1.3.1 객체지향 데이터 모델의 기본 개념

객체를 그 관리 대상으로 하는 데이터베이스 시스템을 객체지향 데이터베이스 시스템이라 한다면, 객체지향 데이터 모델이란 데이터를 모델링하기 위해 유용한 객체지향 개념들을 잘 모아서 정의해 높은 집합이이라 할 수 있다. 그러므로 객체지향 데이터 모델링은 객체지향 시스템의 개발 과정에서 나타나는 객체 모델링과 같다. 객체(objects)가 이미 자기 특유의 속성과 오퍼레이션을 캡슐화한 식별성을 가진 독립체이고, 의미 데이터 모델에서 요구되는 많은 의미적 측면들을 객체지향 개념이 지원하고 있다.

객체스키마는 데이터 속성, 데이터에 대한 제약조건, 함수적 속성을 포함하는 특정한 타입의 객체 속성을 기술한다. 또한 객체 지향시스템은 그 내재적인 특성상 매우 바람직한 방향으로 시스템 개발과정을 이끌어 가게 되는데 그 내재적인 특성은 데이터 추상화, 객체 캡슐화, 계층 그리고 다형성 등을 의미한다. 이를 통해서 소프트웨어 엔지니어링에서 강하게 요구되는 추상화, 모듈성, 응집력, 정보은닉과 같은 원칙들이 자연스럽게 지원된

다. 객체지향데이터모델과 계층형 데이터 모델이 유사하다는 지적들이 많다. 실제로 데이터모델 사이에는 적어도 두 가지 유사성이 있는 것으로 소개되고 있는데, 그 하나는 객체지향 데이터 모델에서의 객체의 중첩된 구조와 계층형(또는 네트워크)에서의 레코드의 중첩된 구조이다. 또 하나는 객체지향 데이터베이스의 객체식별자의 계층형의 레코드 포인터다. 여기에 대해서는 객체지향 데이터 모델에서의 객체 중첩은 순환을 가질 수 있는 반면, 계층형에서 레코드의 순환을 표현하기 위해서는 스키마에 인위적인 레코드형을 만들어 처리해야만 한다. 또한 객체식별자는 논리적인 포인터로 이 식별자는 객체가 삭제되더라도 재사용이 되지 않아 참조 무결성을 강요하기 위해서 사용되는 반면, 레코드에 대한 포인터는 물리적인 포인터로 재사용이 이루어지므로 참조 무결성 검사를 위해서는 사용될 수 없다. 두 가지 유사성을 제외하고는 두 모델을 분명히 구분해주는 차이점들이 있는데, 그것은 객체지향 데이터 모델에서는 클래스 계층, 계승과 오퍼레이션 같은 개념들을 지원한다. 그러나 계층형(혹은 네트워크)데이터 모델에서는 이러한 개념들이 없다. 또한 객체지향 데이터 모델은 구조화된 추상화를 통해 객체의 의미적 측면들을 내포한다는 점에서 의미 데이터 모델과도 유사성이 있는 것으로 지적될 수 있다. 그러나 근본적으로 의미 데이터 모델은 객체의 구조적인 측면을 요약하는 반면, 객체지향 모델은 객체의 행동적 측면을 요약한다.

이는 객체지향 방법론이 다른 아닌 프로그래밍 분야에 출발한 것에서 쉽게 이해될 수 있다. 객체지향 방법론은 뚜렷한 세 가지 특징을 나타내는데, 그 첫째가 객체클래스(또는 타입)의 명시적인 표현이다. 즉 객체는 자신의 값에 의해서 식별되는 것이 아니라 대리자에 의해서 식별된다. 둘째는 객체 내에 메소드(혹은 오퍼레이션)가 캡슐화 된다는 것이다. 사용자는 메소드의 구체적인 구현에 대해서는 관심을 가질 필요가 없다. 세 번째는 그러

한 메소드가 한 클래스에서 다른 클래스로 계승 된다는 것이다.

이러한 객체지향 모델과 의미 데이터 모델간의 주요한 차이는 객체지향모델은 의미 모델의 풍부한 타입 구조체를 전형적으로 구현하지 못한다는 것과 메소드의 계승은 의미 데이터 모델에서의 속성의 계승과는 분명 다른 것이다. 대체적으로 의미 데이터 모델은 메소드라는 개념의 지원에 있어서 객체지향 모델에 미치지 못하지만, 모형화 측면에서 객체지향 데이터 모델은 의미 데이터 모델의 부분집합으로 볼 수 있다. 그러나 객체지향 접근법은 개체-릴레이션 모델이나 SDM(semantic database model)과 의미 데이터 모델에 대해서도 쉽게 연계(correlated)될 뿐 아니라 유지보수성, 확장성, 그리고 재사용성 등 소프트웨어 분야에서의 확인된 이점이 뛰어나기 때문에 데이터나 데이터베이스의 관계를 모형화 하는데 더욱 적합하다.

2.1.3.2 데이터 모델의 비교(선행연구)

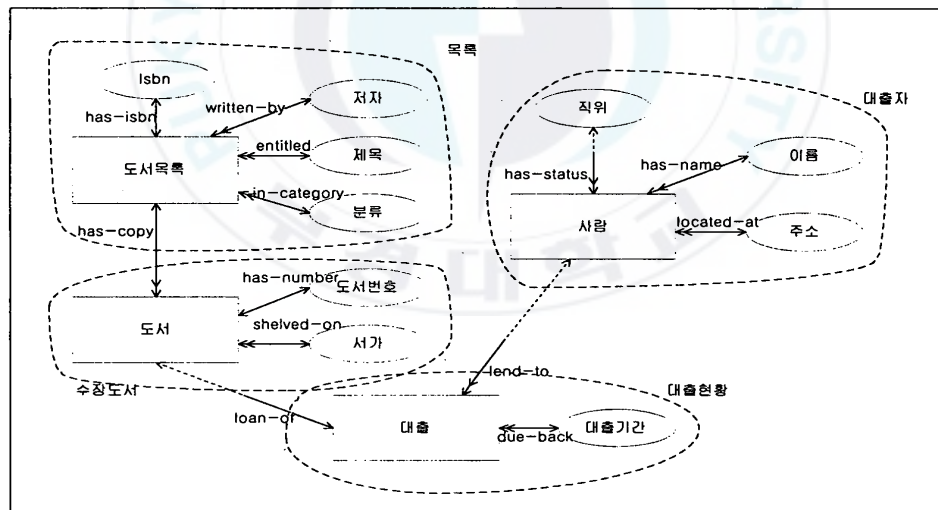
데이터베이스 기술은 세 가지의 주요한 개념에 바탕을 두는데, 그것은 바로 실체(entities), 속성(attributes) 그리고 관계(relationship)을 말한다. 따라서 데이터베이스에 대한 연구의 주된 과제는 이들 세 개념을 어떻게 표현 할 것인가 하는 것이었다. 한 개체는 또 하나의 개체와 연관되기 마련이고, 또 자신만의 속성과 관계를 맺게 된다. 객체지향 데이터베이스에서는 개체와 속성이 모두 하나의 객체(objects)로 간주된다. 또한 개체와 개체간의 혹은 개체와 속성간의 관계도 하나의 독립된 개체로 간주된다.

Oxborrow는 실세계의 개체가 데이터 모형화를 통해서 어떻게 표현되는가를 고찰하였는데, 이는 관계형 데이터 모델과 의미 혹은 객체지향 데이터 모델의 차이점을 우리에게 잘 보여준다. 개념적 모형화를 위한 표기법은 개체를 직사각형으로, 속성을 타원으로 나타내는 등 E-R 모델의 표기와 유사하다. 단 다중 연관관계(cardinality)를 점선과 화살표를 표기하고 있

는데, 예를 들어 다음의 표기는

사람<----->> 대출

한 사람이 여러 번 대출을 할 수도 혹은 전혀 하지 않을 수도 있지만, 대출은 반드시 그리고 항상 한 사람에게만 이루어질 수 있다는 의미를 가진다. 즉 대출 개체에 대해 사람 개체는 일대다의 선택적(optional) 관계를 가지는 반면 대출 개체는 다대일의 강제적 관계를 가진다. 도서관 대출 시스템 개념적 모델이 [그림2-2]와 같을 때, 단순화된 시스템에서 확인되는 개체는 직사각형으로 표현된 도서목록, 도서 대출 그리고 사람이다. 각 개체는 자식들만의 속성들과 관련을 가진다. 점선을 이용하여 집단화한 것이 개체 집합인데, 이는 개체를 관련되는 모든 속성들과 함께 표현하는 전통적인 데이터베이스 관리 시스템에서의 레코드와 유사하다.



[그림2-2] 도서관 대출 시스템의 개념적 모델

출처: E. A. Oxborrow, "An Object-Oriented Approach to Distributed Database Management Systems," Database Technology, Vol. 3, No. 1, 1990, pp. 15.

관계형 데이터 모델의 릴레이션으로 [그림2-2]의 개념적 모델을 표현하면 함께 사용함으로써 각 릴레이션간의 관계를 묵시적으로 표현하게 된다. 그러한 제약들은 바로 주키와 복합키 그리고 외래키의 사용이나 참조 무결성의 개념을 의미한다. 즉 [표2-1]은 “한 릴레이션 내에서 각 튜프들은 유일무이하다”거나 “모든 릴레이션은 적어도 하나의 속성을 가진다”와 같은 고유한 제약조건에 의해 테이블 형태의 릴레이션을 구성했다. Isbn이나 도서번호와 같은 주키를 통해서 “릴레이션 목록내의 어떤 두 개의 튜플도 속성 Isbn의 값을 같은 값으로 가질 수 없다” 제약을 구현하게 된다. 또한 “각 속성은 원자적이다”라는 고유한 제약을 “한 릴레이션의 어떤 두 튜플도 속성A에 대해서는 같은 값을 가질 수 없다”라는 명시적 제약과 함께 적용함으로써 한 릴레이션 내에서 주키를 제외한 모든 속성이 주키에 대해서 함수적으로 완전히 종속하게 된다.

이 때 그림 [2-2]에 나타난 개체간의 여러 관계들이 복합키를 이용하여 내부적으로 표현되는데, 예를 들면 개체 도서와 개체 대출간의 loan_of 관계와 개체 대출과 개체 사람간의 lent-to 관계를 릴레이션이 대출현황의 복합키 도서번호, 이름, 주소에 의해 내부적으로 표현되고 있다. 그러나 사실상 [표2-1]에는 표[2-2]와 같은 종속관계가 숨어 있다.

[표2-1]과 [표2-2]를 합하여 [표2-3]고 같이 나타낼 수도 있다. 그러나 [표2-3]도 그림[2-2]와 비교해 볼 때, 결코 그 의미를 충분히 표현했다고는 볼 수 없다. 또한 [그림2-2]자체가 각 개체가 가지는 특유의 오퍼레이션은 포함하지 않고 있으므로, 만약 필요한 오퍼레이션이 정의될 경우에는 그에 포함하지 않고 있으므로, 만약 필요한 오퍼레이션이 정의될 경우에는 그에 대한 배려도 있어야만 할 것이다. 관계형 데이터 모델에서는 그러한 경우에 대해서는 전혀 대비가 되어있지 않다.

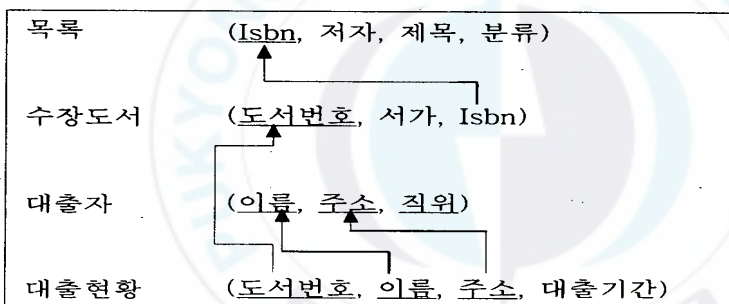
[표2-1] 도서대출시스템의 관계형 모델

목록	(<u>Isbn</u> , 저자, 제목, 분류)
수장도서	(<u>도서번호</u> , 서가, Isbn)
대출자	(<u>이름</u> , 주소, 직위)
대출현황	(<u>도서번호</u> , <u>이름</u> , 주소, 대출기간)

[표2-2] 도서 대출 시스템의 종속 관계

Isbn[수장도서]	⊆ Isbn[목록]
도서번호[대출현황]	⊆ 도서번호[수장도서]
이름[대출현황]	⊆ 이름[대출자]
주소[대출현황]	⊆ 주소[대출자]

[표2-3] 도서 대출 시스템 관계형 모델의 의미 표현



객체지향 방법론이 차세대 데이터베이스 기술로 각광받게 되는데는 최소한 두가지 이유가 있다. 첫째, 객체지향 방법론은 관계형 데이터 모델을 포함하는 데이터 모델의 기초를 제공할 수 있다. 객체 모델은 관계형 모델의 제한된 데이터형 뿐만이 아니라, 사용자가 정의한 데이터형을 지원하고 나아가 데이터와 오퍼레이션을 캡슐화 할수 있다. 둘째, 캡슐화와 계승(재사용)을 통해서 복잡한 소프트웨어 시스템의 개발 및 개선의 난점들을 줄여 준다는 것이다.

객체 모델로 [그림2-2]를 표현한다면, 객체 모델링 절차에 입각하여 객체를 식별하는 작업부터 시작해야 하는데 우선적으로 도서목록, 도서, 사람,

그리고 대출 등의 실체가 객체로 정의된다. 여기서 각 실체에 연관되는 속성들도 하나의 객체로 정의되고, 각 객체들간의 관계도 또한 객체로 정의된다. 예를들어 도서목록 집합의 한 멤버가 제목 집합의 한 멤버에 대해서 가지는 has-title이라는 데이터 속성과 제목 집합의 한 멤버가 도서목록 집합의 부분집합에 대해서 가지는 is-title-of의 데이터 속성을 가진다. 관계 객체는 다른 객체의 데이터 속성으로도 포함될 수 있다.

이러한 개체 객체나 속성객체 또는 관계객체 복합 데이터 모델을 구성하기 위한 기반이 된다. 복합 데이터 모델, 즉 복합객체란 기본객체(클래스)로부터 유도되어 구성된 객체(클래스)를 말한다. 개체지향 데이터 모델은 위에서 설명한 기초 객체들로부터 유도된 복합 객체의 형태로 나타나게 된다.

2.1.3.3 관계형 데이터 모델의 확장

관계형 데이터 모델이 가지는 한계점들을 극복하기 위해서 관계형 데이터 모델을 확장하고자 하는 연구가 많이 진행되고 있다. 그 확장의 방법 중 하나가 객체지향개념을 지원할 수 있게 하는 것이다.

첫 번째 확장은 릴레이션의 열값이 시스템에서 정의한 자료형뿐만 아니라 사용자가 정의한 릴레이션의 튜플이 될 수 있도록 허용함으로써 객체지향 개념을 반영한다. 이는 사용자가 정의한 임의의 릴레이션을 다른 릴레이션의 열값의 범주로 지정할 수 있다는 것이다. 릴레이션의 열이 다른 릴레이션의 튜플이 될 수 있도록 허용함으로써 객체지향 개념을 반영한다 이는 사용자가 정의한 임의의 릴레이션이 다른 릴레이션 튜플을 가지게 되면 중첩 릴레이션이 발생한다. 즉 릴레이션의 행/열 엔트리의 값이 다른 릴레이션의 튜플을 가질 수 있다.

두 번째 확장은 객체지향 개념인 캡슐화를 지원할 수 있게 하는 것이다.

캡슐화를 자료를 조작하기 위한 프로그램과 자료를 결합하는 것인데, 이는 한 릴레이션에 속하는 튜플의 열에 대한 연산을 수행하는 처리절차를 그 릴레이션에 부착시킴으로써 가능하다. 이를 통해 한 릴레이션은 이제 그 릴레이션의 튜플의 상태와 작동을 캡슐화한다. 상태는 열값들의 집합이고 동작은 그 열값에 대하여 연산을 수행하는 어떤 처리절차도 만들어 사용할 수 있게 된다.

세 번째 확장은 계층의 개념을 지원하는 것이다. 사용자는 데이터베이스에 있는 모든 릴레이션을 계층으로 구상할 수도 있을 것이다. 릴레이션의 계층은 시스템에 정의된 루트를 갖는 비순환 방향 그래프이다. 일반화와 구체화 관계는 자식과 그의 부모 릴레이션의 사이에 성립된다. 릴레이션의 계층은 관계형 모델의 독립적인 릴레이션들의 집합에 비해 두 가지 장점이 있는데, 그것은 릴레이션 계층을 통해 사용자가 이미 존재하는 릴레이션의 자식으로 새로운 릴레이션을 생성할 수 있다는 것과 두 릴레이션의 관계를 IS_A로 유지할 수 있다는 것이다.

이러한 확장의 결과는 전체적으로 데이터 모델링을 용이하게 하고, 질의성능을 증가시킨다는 면에서 중요하다. 중첩 릴레이션과 열에 집합을 허용하는 확장은 관계형 데이터베이스의 사용자들을 성가시게 하였던 일을 제거한다. 처리절차와 릴레이션 계층의 확장은 데이터 모델링과 응용 프로그램에 새롭고도 중요한 가능성을 제시한다. 더욱이 중첩 릴레이션과 릴레이션 계층의 확장은 객체지향 프로그램 언어의 강력한 자료형 기능을 반영한다. 이렇게 관계형 데이터 모델이 객체지향 개념의 핵심적인 측면들을 지원하게 확장시킨다면 그 결과는 다름 아닌 객체지향 데이터 모델이 되는 셈이다.

2.2 객체지향 방법론과 데이터 모델 도출

2.2.1 객체지향 방법론

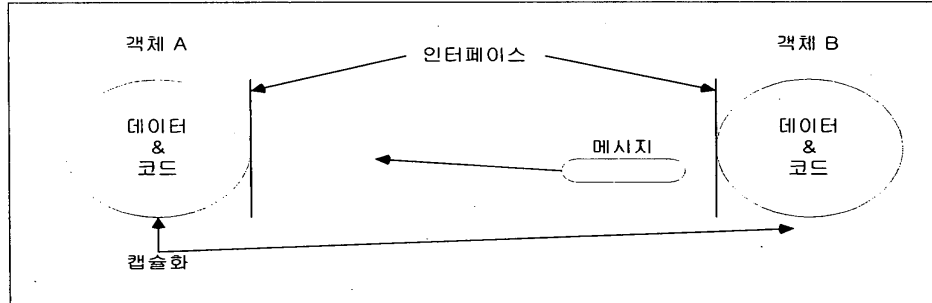
2.2.1.1 객체지향 시스템

객체지향(object orientation)이란 실세계의 개체와 개체간의 관계를 객체라고 하는 자기충족성(self-sufficiency)을 가진 구조체를 중심으로 바라보겠다는 것이다. 객체지향 접근법의 본질은 시스템 구조화를 위한 두 가지 원칙을 관철하는 것인데, 그 두 가지 원칙은 객체 내에 데이터와 관련 오퍼레이션을 함께 저장하겠다는 캡슐화(encapsulation)와 객체 클래스간에 공통요인들(commonalities)을 공유하겠다는 계승(inheritance)을 말한다. 객체지향과 관련된 용어들을 정리하면 [표 2-4]와 같다.

이상과 같은 용어에 대한 정리를 기반으로 객체와 객체간의 상호작용을 도식해보면 [그림 2-3]과 같이 표현될 수 있다.

2.2.1.2 객체지향 접근법

객체지향 시스템은 주어진 목표를 달성하기 위해 메시지 전달을 통해 상호작용하고 커뮤니케이션 하는 객체들의 집합으로 볼 수 있고 객체지향 시스템을 개발해 나가는 과정은 절차중심의 시스템 개발과정과 다를 수 밖에 없다. [2] 시스템 개발 과정에 있어서 기존의 절차중심 접근법은 시스템을 process로 본다. 관련된 데이터 항목을 레코드 및 릴레이션과 같은 구조체로 묶고, 데이터를 조작하는데 필요한 함수 또는 처리 절차를 작성한다. 함수들은 데이터와는 별개로 해야 할 일의 순서에 따라 호출되며, 함수에서는 해야 할 일에 따라 데이터를 조작하게 된다.



[그림2-3] 캡슐화와 메시지전달

출처: E. Nahouraii & F. Petry, *Object-Oriented Databases*, IEEE Computer Society Press, Chelsea(Michigan), 1991, pp. 3.에서 수정.

[표 2-4] 객체지향 관련 용어

클래스	동일한 구조와 행동을 공유하는 객체들의 집합에 대한 추상적인 정의 혹은 금형(template)
캡슐화	캡슐(객체)내에 데이터 구조와 그 구조들에 접근할 수 있는 모든 오퍼레이션을 묶어둠으로써 그 오퍼레이션을 통하지 않은 다른 수단으로는 데이터에 직접 접근하는 것을 방지함.
정보은닉	한 모듈의 내부적 구조가 여타의 모듈로부터 감추어진 상태로 존재해야함을 의미하는 설계 원칙으로 모듈의 내부적 표현에 영향을 줄 수 있는 여하한 변화로부터 여타의 모듈들을 격리함
계승	객체가 서로 다른 그러나 관련있는 클래스로부터 공통의 특성들을 공유할 수 있도록 그 공통의 특성들을 상위 수준의 클래스에 위치시키고 각 클래스들과 연결을 형성하는 메커니즘
메시지전달	객체간의 유일한 커뮤니케이션을 위해 한 객체로부터 다른 객체로 명시적인 메시지를 보내는 방법인데, 여기서 메시지는 객체식별자와 메소드명, 매개변수들을 포함함
메소드	객체내의 데이터에 대해서 원자적이고 결속력을 가진 오퍼레이션을 수행하는 객체에 연관된 작은 양의 프로그램
객체	실세계의 객체를 추상화한 것으로 실세계 객체의 속성과 행동에 대응하는 변수와 메소드의 집합을 캡슐화함
다형성	별개의 그러나 유사한 클래스에서 기인하는 객체에 대해 전달된 동일한 메시지가 공통적인 액션을 호출하게끔 하는 프로그래밍 메커니즘
변수	한 객체에 관련된 데이터 아이템(속성)

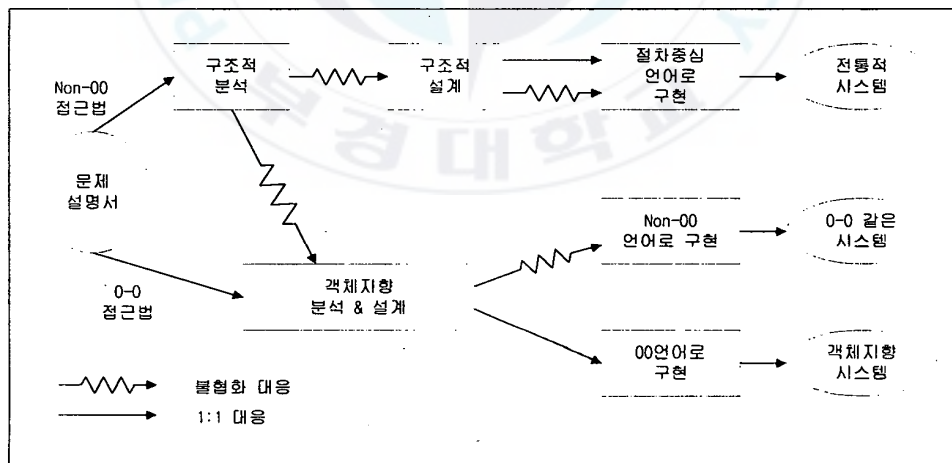
그러나 프로그램이 존재하는 이유는 데이터 그 자체에 있다는 것을 생각할 때 절차적 접근은 문제점을 가질 수 밖에 없다. 이와는 달리 객체지향 접근법의 주개념은 절차(process)보다는 개체이다. 이는 곧 입력물의 출력물로써의 변환이 아니라 그 개체의 기초를 이루는 내용(concept)을 강조하는 것이다. 그 내용은 다른 아닌 개체를 정의하는 데이터 구조와 시간의 경과와 함께 변화되는 처리절차(process)의 상태, 또는 반복적 수행 주기에 걸쳐 불변하는 처리절차의 일면 등이 된다.

이와 같이 객체지향 접근법에서는 시스템을 다른 객체에 대해 반응하기도 하고, 다른 객체에 대해 어떤 행위를 요구하기도 하는 자기충족성을 가진 개체(objects)의 집합으로 본다. 시스템 설계자는 일련의 데이터와 그 데이터에 필요한 오퍼레이션들을 함께 위치시킴으로써 캡슐화를 달성한다. 실제로 어떤 주어진 데이터 구조에 대해 민감한 오퍼레이션의 수는 한정적일 수 밖에 없다. 따라서 그 데이터와 오퍼레이션만을 한 곳에 모아두겠다는 것은 논리적이다.

계승도 실세계의 계승과 같은 개념이 된다. 부모로부터 특정한 속성과 행동양식을 유전적으로 계승받는 것처럼 객체지향 시스템 내에서도 객체의 클래스는 부모 클래스로부터 속성과 메소드를 계승하는 것이다. 이를 통해서 시스템 개발자는 시스템 모듈 즉, 객체 클래스이 계층의 적당한 수전에 각 오퍼레이션들을 위치시킴으로써 중복되는 코딩작업을 피할 수 있게 된다. 시스템 설계자는 이러한 캡슐화와 계승을 통해서 추상화, 모듈화 그리고 재사용성과 같은 소프트웨어 공학의 목표들을 달성할 수 있다.

객체지향 시스템은 객체지향 접근법에 의한 시스템 분석 및 설계, 개발의 산물이다. 객체지향 시스템은 개발과정에서부터 전통적인 절차중심

의 접근법과는 다를 수 밖에 없다. 구조적 분석·설계의 과정을 통한 시스템 개발과정은 이미 언급한 바와 같이 내재적인 문제점으로 인해 변칙(anomaly)이 끼여들 가능성을 늘 보유하고 있다. 객체지향 접근법은 시스템 개발자나 사용자가 느끼는 그대로의 실세계 객체를 최대한 변칙 없이 시스템 내의 객체로 이식하려는 노력의 산물인 만큼 현실 세계의 문제 영역을 컴퓨터 시스템내의 해결 영역으로 변환하려는 소프트웨어 엔지니어링의 목적에 보다 부합하는 방법론인 것이다. 객체지향 접근법에 있어 주목할 또 하나의 특징은 시스템 분석단계와 시스템 설계 단계가 명확히 구분되지 안혹 같이 이루어진다는 것이다. [3] 이는 분석·설계된 시스템을 그 즉시로 구현해 보여줌으로써 사용자의 참여도를 높이는 원형(prototype) 개발법과 같이 사용됨을 의미한다. 객체 지향적 분석이 이루어지지 않은 경우에도 시스템 설계 과정에서 객체지향 접근법을 도입한다면 객체지향 방법론이 제공하는 다양한 이점들을 얻을 수도 있을 것이다. [그림 2-4] 는 이러한 관계들을 잘 보여주고 있다.



[그림2-4] 시스템 개발 과정

출처: B. Henderson-Sellers & J. M. Edwards, "The Object-Oriented System Life Cycle," Communications of the ACM, Vol. 33, No. 9, 1990, pp. 147.에서 수정.

2.2.1.3 객체기반 시스템

객체지향 설계 기법은 사용자가 객체지향 언어를 사용하고자 할 때 큰 도움이 될 수 있다는 것은 분명하다. 객체지향 언어란 최소한 객체, 계승, 다형성(polymorphism) 등의 새로운 특성들을 지원하는 언어를 말한다. 객체는 데이터와 처리절차를 캡슐화하므로 처리절차(procedure) 내의 코드는 데이터를 직접 참조하게 된다. 시스템의 나머지 부분들은 멤버함수를 통해 파라미터를 넘겨줌으로써 메시지만을 보내어 결과를 통보 받을 뿐이다. 여기까지의 결과는 바로 추상적인 데이터 형 (abstract data type : ADT) 즉 객체인 것이다.

객체 개념의 도입은 상당히 이상적이긴 하지만, 실제적인 유지 문제에 대해서는 해결책을 가지고 있지 않다. 시스템을 개선하거나 사용자가 약간의 수정을 가하고 싶은 경우, 이미 존재하는 객체에 새로운 변형 (variants)을 추가함으로써 개선 및 수정을 가하게 된다. 이 때 프로그램이 추상 데이터형의 개념만으로 설계되어 있다면 코드를 호출하는 부분은 단지 그 객체들 중 하나만을 참조할 수 있게 될 것이다. 코드를 호출한다는 것은 호출하려는 객체의 종류를 알아야만 되므로 개선 및 수정으로 인한 변화는 변동사항이 있는 객체에만 국한되지 않게 된다. 이는 객체지향 방법론이 강점으로 주장하는 유지용이성과는 거리가 먼 것이다. 이러한 문제를 해결해 주는 것이 바로 계승 개념을 지원하는 객체 분류 구조와 연결되어 있는 다형성인 것이다. 동일한 하나의 메시지가 서로 다른 객체에서 각각 자기 방식대로 해석될 수 있겠다. 이것이 바로 객체지향(object-oriented)과 객체기반(object-based)을 구분하는 기준이 된다. 객체기반 언어라는 것은 추상 데이터형을 보유하고, 객체지향 언어는 거기에 계승과 다형성을 지원한다는 것이다. 따라서 객체기

반 설계는 추상 데이터형을 사용하는 설계 기법이고, 객체지향설계는 거기에 계승과 다형성 개념을 추가로 사용하는 설계 기법인 것이다.

두 개념에 대해서 Korson과 McGregor도 유사한 견해를 밝히고 있는데, 절차 중심 설계 방법은 문제를 해결하기 위해서 완료되어야 할 임무를 구체화하기 위해 기능적 분해(functional decomposition)를 활용하고, 객체기반 접근법은 절차적 접근보다는 데이터 명세에 보다 관심을 기울이되 아직도 시스템 구조를 개발하기 위해서 기능적 분해를 활용한다는 것이다. 객체지향 접근법은 여기서 객체와 객체간의 관계를 시스템 구조의 본원적 구성요소로 보고 데이터에 강조점을 더 둔다는 것이다. 결국 추상 데이터형(ADT)이 개념적인 정보를 포착하기 위해 사용되는 객체기반 방법론의 기법이라면, 클래스라는 것은 객체지향 방법론의 개념적 모형화 도구 인 것이다.

객체지향 방법론과 객체지향 시스템이 사용자 지원의 측면에서 탁월한 장점이 있는 것은 주지의 사실이나 이미 사용중인 전통적인 시스템들을 모두 폐기할 수는 없는 것이다. 더욱이 상용화된 객체지향 시스템들 중 일부는(예, 객체지향 데이터베이스 시스템) 성능과 사용자 신뢰도에 있어서 전통적 시스템(예, 관계형 데이터베이스 시스템)에 미치지 못하는 경우도 있다. 실제로 대부분의 데이터베이스 시스템 사용자들은 객체지향 프로그래밍과 객체지향 데이터베이스 시스템의 강점을 인식하고는 있으나, 객체지향 데이터베이스 관리시스템의 불완전성을 이유로 사용을 기피하고 있다. 따라서 전통적인 시스템을 살리면서 객체를 기반으로 체계화하여 접근하고 사용하려는 시도는 객체지향 개념의 일부를 희생하면서 요구되는 한계점의 개선에 대해 만족할 만한 성과를 제공할 수도 있다.

2.2.2 복합 개체

복합 개체(composite or complex objects)란 다른 여러 클래스로부터 생성된 이질 객체의 모임인 하나의 논리적인 단위이다. 객체지향 데이터 모델에서 각 객체의 속성은 임의의 클래스를 그 영역으로 할 수 있다. 따라서 이 개념만으로도 각 객체들 사이의 구성 관계를 나타내기에는 충분하다. Part-of 관계는 하나의 객체와 그것이 참조하는 다른 객체들간의 통합관계를 표현하기 위해 사용되는 관계로서 이를 이용하여 복합객체를 명세할 수 있다. 객체지향 시스템에서 객체는 속성의 집합으로 표현되고, 각 속성값은 그 속성의 정의역의 한 인스턴스가 되는데 또 다른 객체이다. 예를 들어 어떤 속성의 정의역이 기본클래스로 정의되어 있으면, 반드시 정수나 문자열의 집합이 그 해당 속성에 저장되는 값이 될 수 있다. 속성의 정의역이 기본클래스가 아니면 해당 속성은 그 속성의 해당 범위에 속하는 한 인스턴스에 대한 객체식별자를 값으로 갖는다. 객체지향 데이터베이스 복합 개체는 여러 개의 기본 또는 사용자 정의 타입으로 구성된 객체이다. 합성, 집합, 참조는 객체-관계 데이터베이스관리시스템에서 복합 타입을 생성하는 기본적인 블록들이다. 합성 타입을 생성하려면 타입의 이름과 함께 구성 요소의 이름과 데이터 타입을 명시하여야 하며, 합성 타입의 요소에 따른 합성 타입을 사용할 수 있다. 객체-관계 데이터베이스관리시스템에서는 시스템을 새로운 기본 데이터 타입과 함수의 정의 기능, 그리고 집합, 참조, 합성의 타입 생성자를 사용한 복합객체와 함수의 정의 기능을 사용하여 확장 할 수 있게 한다. 그것은 자연스러움을 제공하고 캡슐화를 가능하게 하며 OID를 사용하여 공간 사용에 따른 부담을 줄인다.

복합 객체의 하위 객체들도 다시 새로운 복합 객체가 될 수 있다. 일반적으로 복합 객체는 구성 객체들과 그 관계를 기술함으로써 정의되며, 복합 객체의 구체화(instantiation)는 그 구성 객체들을 모두 구체화함으로써 이루어진다. 이 때 구성 객체가 다시 복합 객체가 될 수 있으므로, 복합 객체의 구체화는 재귀적(recursive)으로 발생한다. 복합 객체 개념의 도입은 객체간에 새로운 종속 관계를 발생시킨다. 예를 들면, 자동차 없이 차체만 존재한다든지, 교실 창문 객체가 교실 객체없이 존재하는 것은 아무런 의미가 없으며, 차체나 교실 창문 객체는 각각 자동차와 교실 객체가 존재하는 경우에만 이 객체들에 종속하여 존재하게 된다. 이렇듯 객체의 존재가 그 객체를 소유하는 다른 객체에 달려있을 때 이를 종속 객체(dependent object)라 한다.

2.2.3 객체지향 데이터 모델의 의의

객체지향 데이터 모델은 물론 객체지향 데이터베이스 기술의 개념적 수준의 구조를 나타내는 모델이 된다. 그러나 객체지향 데이터베이스 기술은 아직 완전한 범용시스템으로서는 인정받지 못하고 있다. 그것은 데이터만을 잘 관리해주던 관계형 데이터베이스 시스템과 같은 기존의 상용데이터베이스 시스템들보다 객체들간의 관계를 포함하는 많은 의미적 측면들과 데이터 함께 캡슐화된 오퍼레이션들을 지속적으로(consistently) 저장하고, 수많은 인스턴스에 대한 질의들에 효과적으로 반응하도록 지원하는 것이 기술적으로 어려운데서 기인한다. 객체지향 데이터 모델을 지원해 주는 데이터베이스 기술과 그에 대한 기대와 한계, 그리고 그러한 문제점들의 극복을 위한 노력을 아래에 기술한다.

2.2.3.1 객체지향 데이터베이스 기술

객체지향 데이터베이스 시스템은 많은 양의 데이터를 관리하고 지속성 데이터와 트랜잭션의 무결성 및 동시성, 보안성 유지와 시스템 회복 체계, 접근 언어를 제공하는 데이터베이스로서의 기본적인 능력에 더하여 추상화, 계승 그리고 객체 식별성(object identity)을 제공하는 데이터베이스 시스템을 말한다. 또한 버전 제어 기능까지 가지는 시스템으로 기대할 수도 있다. 다시 말하면, 객체지향 데이터베이스는 객체(object)를 생성하고, 저장·유지하고, 검색하고, 추출하는 관리 시스템인 것이다. 이를 위해서는 객체를 조직화하여 클래스를 생성하고, 객체를 생성하고(instantiation), 하위 클래스가 상위 클래스의 속성과 메소드를 계승받을 수 있도록 클래스를 조직하기 위해 계승 계층을 형성하고, 특정 객체에 접근하기 위해 메소드를 호출하는 등의 기능을 제공해야만 한다. 관계형 데이터베이스에 빗대어 객체지향 데이터베이스를 정의해 본다면, 두 시스템간의 본질적인 차이는 바로 관리대상으로 하는 자료 모델이 다르다는 것이다. 즉, 자료가 어떻게 표현되느냐 하는 것이다. 관계형 데이터베이스에서 속성과 릴레이션으로 표현된 자료가 객체지향 데이터베이스에서는 속성, 메소드, 클래스, 클래스계층으로 표현된다.

2.2.3.2 객체지향 데이터베이스에 대한 기대와 한계

관계형 데이터베이스의 한계점들을 객체지향 데이터베이스가 상당부분 보완해 주리라 기대할 수 있다. 그러한 기대를 가능케 하는 것은 객체지향 개념을 통해서 데이터베이스 설계와 프로그램을 재사용할 수 있게 된다는 점과 캡슐화와 계승이라는 객체지향 개념에 내재되어 있는 강력한 자료형 기능에서 기인한다.

객체지향 개념은 원래 복잡한 소프트웨어 시스템이나 각종 설계를 관리학 위하여 설계되었다. 캡슐화와 계층을 통하여 자료구조(즉, 데이터베이스 설계)와 프로그램을 재사용하여 보다 복잡한 데이터베이스와 프로그램을 구축하는 발판이 되었다. 이는 바로 과거 30년 동안 파일 시스템에서 관계형 데이터베이스에 이르기까지 자료관리 기술이 추구한 목표였다. 객체지향 데이터베이스는 매우 복잡한 데이터베이스를 설계하고 관리하고 있어 어려움을 줄일 수 있는 잠재성을 갖고 있다.

그러나 현존하는 대부분의 객체지향 데이터베이스 시스템은 지속성을 지원하는 저장 시스템으로서나 범용 데이터베이스 시스템으로서의 한계를 가지고 있다. 따라서 관계형 데이터베이스와 같이 상용화되기 위해서는 다음과 같은 개선이 뒤따라야 할 것이다.

- ① 객체지향 데이터베이스에 사용할 수 있는 질의어를 설계해야 한다.
- ② 질의어를 처리하기 위해 구문 처리기(parser)를 구현해야 한다
- ③ 질의 최적기를 추가해야 한다.
- ④ 복잡한 질의를 처리하기 위해 정렬병합결합, 질의 평가, 중첩 질의 처리등과 같은 다양한 알고리즘을 구현해야 한다.
- ⑤ 데이터베이스에 자료를 효율적으로 저장 또는 추출할 수 있는 접근 기법을 구현해야 한다.
- ⑥ 데이터베이스에 대한 통계를 제공하는 카탈로그를 개발해야 하며, 질의 처리를 위해서 이러한 통계 자료를 관리하는 카탈로그 관리기를 구현해야 한다
- ⑦ 동시성 제어 및 회복 기법과 통합되어 자료접근 기법이 구현되어야 한다.
- ⑧ 질의 처리 알고리즘도 데이터베이스 시스템의 동시성 제어 및 회복 기법과 통합되어 구현되어야 한다.

⑨ 질의 처리를 위해 작업영역 관리기가 수정되어야 한다.

이외에도 사용자가 스키마를 변경할 수 있도록 하기 위해서 몇가지 기능들이 추가되어야 한다.

객체지향 데이터베이스의 개발은 기술적인 어려움뿐 아니라 객체지향 개념에서 유래된 어려움도 있다. 따라서 대부분의 객체지향 데이터베이스는 데이터베이스 시스템이라기 보다는 객체지향 프로그래밍 언어에 지속성을 지원하는 저장 시스템이라고 할 수 있다. 따라서 객체지향 기법과 기존의 관계형 기법을 통합하려는 노력이 새롭게 일어나고 있는 것이다.

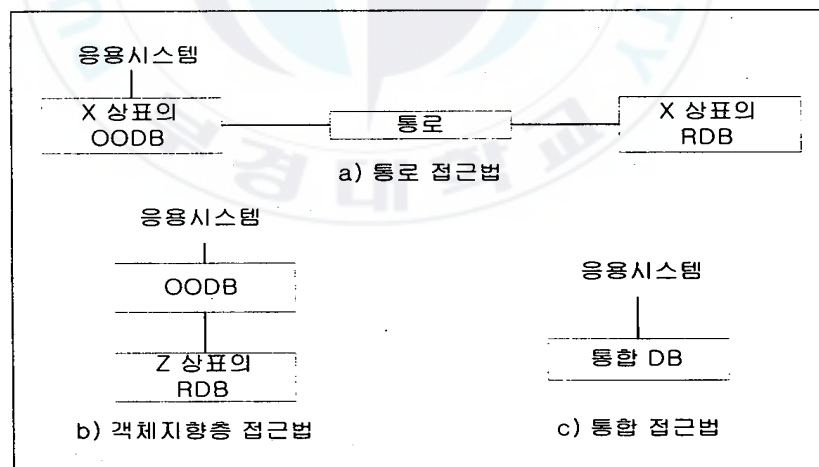
2.2.3.3 관계형과 객체지향 기법의 통합

관계형 데이터베이스의 결함을 해결할 대안으로서의 객체지향 데이터베이스에 대한 기대는 널리 퍼져 있다. 그러나 데이터베이스 시장에서 객체지향 데이터베이스가 아직 큰 영향을 미치지 못하는 것은 현재 대부분의 객체지향 데이터베이스가 관계형 데이터베이스가 이미 지원하는 많은 주요 기능을 지원하지 못하기 때문이다. 이러한 현실은 두 기법을 적절히 통합함으로써 양자의 장점을 부각시키려는 노력들이 뒤따르는 이유가 된다.

객체지향 데이터베이스와 관계형 데이터베이스를 결합하는 방법은 크게 세 가지 정도로 분류할 수 있는데, 두 데이터베이스 사이에 통로(gateway)를 두는 방법, 관계형 데이터베이스 위에 객체지향 계층(OO-layer)을 두는 방법, 둘을 하나로 통합하여 단일 시스템으로 구축하는 방법 등이 그것이다. 이들은 [그림 2-5]와 같이 나타낼 수 있다. 통로를 두는 방법에서 객체지향 데이터베이스는 사용자의 질의를 단순히 변환하여 관계형 데이터베이스로 전송하고, 관계형 데이터베이스는 이를 수행하여 그 결과를 다시 객체지향 데이터베이스에 전송한다.

관계형 데이터베이스에서 볼 때 통로는 관계형 데이터베이스의 일반 사용자와 가탄. 객체지향 계층을 두는 방법에서는 사용자가 객체지향 데이터베이스 언어를 이용하여 시스템을 접근한다. 객체지향 계층은 데이터베이스 언어에 나타난 모든 객체지향 측면을 하부의 관계형 데이터베이스와 상호작용할 수 있는 동등한 관계형으로 변환한다. 단일 시스템으로 통합하는 방법은 관계형 데이터베이스의 저장 관리 계층과 자료 관리 계층에 필요한 모든 변환을 하여 객체지향 데이터베이스와 관계형 데이터베이스를 하나의 계층으로 만드는 것이다.

[4] 관계형 데이터베이스를 객체기반으로 관리하겠다는 것도 두 가지 기법을 통합하려는 노력들 중 하나이다. [그림 2-5]에서 알 수 있듯이 연구자나 기업들은 데이터베이스를 관리하기 위하여 관계형 데이터베이스 시스템에다 객체지향 데이터베이스 시스템을 병용하는 방법들을 개발하려고 한다. 이는 분명히 막대한 추가 비용을 발생시킴에도 불구하고



[그림2-5] OODB와 RDB의 통합방식

출처: 김원, “객체지향 데이터베이스 기술”, 정보과학회지, 11권, 2호, 1993, pp. 52.

데이터베이스 관리라는 하나의 목적을 위해 두 가지 시스템을 사용하겠다는 것이다. 이러한 동향에 대해서 우선은 그 만큼 객체지향 시스템이 많은 장점들을 사용자나 개발자에게 제공한다는 것을 읽을 수 있는 반면, 완전한 객체지향 데이터베이스 시스템의 지원을 필요치 않는 경우에 있어서는 막대한 비용을 시스템 구입에 투자하는 것보다 객체기반의 시스템을 추가함으로써 객체를 중심한 시스템 개발과 사용의 이점들을 얻을 수도 있을 것이다.

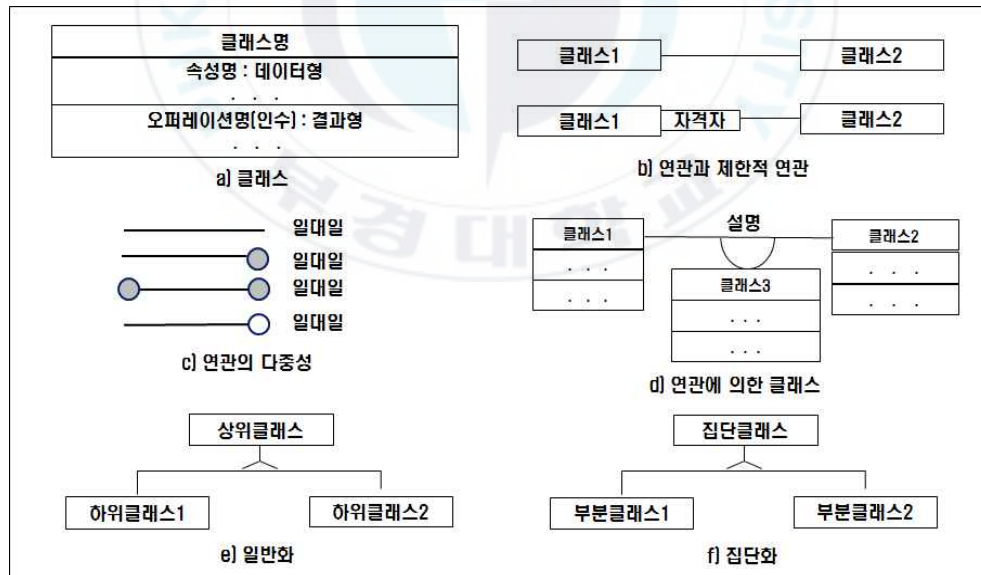
2.3 객체지향 방법론 과 데이터 모델 개발 절차

2.3.1 OMT 기법

객체지향 방법론은 문제 영역에서 객체를 확인하여 모형화하고 이를 계층적으로 수정 보완하여 표현된 객체와 객체간의 관계들을 실제로 구현하는 과정으로 볼 수 있다. 객체를 모델링하는 과정을 지원하기 위해 그래픽기반의 기법들이 많이 소개되었는데, 이들은 모두 시스템 개발의 각 단계를 지원하는데 분석, 설계, 구현의 3단계 전체를 효과적으로 지원하는 기법은 Rumbaugh의 OMT(Object Modeling Technique)인 것으로 평가되고 있다. 본 연구에서도 객체지향 개념을 지원하는 데이터 모델을 도출하기 위해 OMT를 사용하게 될 것이다.

OMT는 세 가지의 모델을 구성함으로써 서로 다른 각도에서 문제 영역을 분석하고 시스템을 설계한다. 핵심이 되는 객체 모델(object model)은 문제 영역에 존재하는 객체들을 확인하여 클래스를 생성하고,

그 클래스간의 정적인 관계를 기술하게 된다. 클래스와 클래스 관계의 기술은 객체 다이어그램이라는 약속된 도시법을 가지는데 클래스, 연관, 일반화, 집단화, 속성, 오퍼레이션 등이 특수한 기호로 표기된다. 또 하나는 시간의 변화에 따라 시스템이 변환되는 과정을 기술하는 동적모델(dynamic model)로, 이는 시스템의 제어와 관계되는 부분을 기술하며 이를 상태 다이어그램(static diagram)을 이용하여 도시한다. 끝으로 시스템 내부에서 데이터가 변환되는 과정을 나타내는 기능모델이 있다. 기능 모델은 구조적 분석에서 사용되는 데이터 흐름도(data flow diagram: DFD)를 이용하여 도시한다. 세 가지 모델 중 핵심이 되는 것은 객체모델이다. 객체지향 방법론에 있어서 객체를 식별하여 정의하고, 그들간의 관계를 정리한다는 것이 핵심이 되는 것은 당연하다. 본 연구의 목적상 요구되는 객체 모델의 표기법은 [그림2-6]과 같다.



[그림2-6] OMT 표기법

2.3.2 객체 모델링 절차

객체를 모델링한다는 것은 그 기저에 적어도 두 가지의 가정이 숨어 있다.

그 한 가지는 데이터 모델이 단순히 선정된 사용 데이터베이스 시스템에 맞는 스키마를 개발하기 위해서지만 그 의미를 가지는 것이 아니라는 것이다. 데이터베이스 설계는 그 데이터베이스를 기반으로 하는 응용 프로그램이 강요해야만 하는 모든 법칙에 대한 목록이 된다는 점에서 장래의 응용 프로그램 사용자에게는 중요한 도구가 된다. 또 한 가지는 객체 모델링은 의미 데이터 모델링과 마찬가지로 개체와 그들간의 관계를 표현한 것이므로, 객체 모델을 완벽하게 도출해 냈다는 것은 훌륭한 데이터 모델을 이미 생성한 격과 같다는 것이다. 데이터 모델을 어떤 데이터베이스 시스템으로 구현 하느냐 하는 것은 시스템분석설계를 통해 개발된 시스템 모델을 어떤 프로그래밍 언어로 구현하느냐 하는 것은 문제이다.

객체 모델링은 실세계 문제 영역으로부터 객체와 클래스를 추출하여 그들간의 관계(연관관계, 집단화 및 일반화 관계 등)를 객체 다이어그램으로 표현하며, 여기에 클래스의 속성과 오퍼레이션을 함께 표현함으로써 시스템의 정적 구조인 객체 모델을 생성하게 된다. 객체 모델을 구축하는 단계는 다음의 여덟 단계로 나누어 볼 수 있다.

- ① 객체와 클래스를 식별한다.
- ② 데이터 사전을 작성한다.
- ③ 객체들간의 연관(association)관계(집단화포함)를 확인한다.
- ④ 객체의 속성과 연결속성(link attributes)을 확인한다.
- ⑤ 일반화(generation)통해 객체 클래스를 조직하고, 단순화한다.

⑥ 접근경로가 유사한 질의에 대해 존재 하는가 검사한다.

⑦ 이상의 과정을 반복하여 모델을 정제한다

⑧ 클래스를 모듈로 집산화 한다

객체 모델을 생성하는 과정을 분석단계와 설계 단계로 엄격히 구분하여 분석단계는 문제 영역(problem domain)에 초점을 두고서 이루어지는 것이고, 설계 단계는 해법 영역(solution domain)을 다루는 것으로 파악하는 견해도 있다. 그러나 이미 언급한 바와 같이 객체지향방법론이 시스템 개발 도구로서 가지는 특징 중에 하나가 분석과 설계가 함께 이루어져 둘 사이의 구분이 거의 없다는 것이 일반적인 관점이다.

2.3.3 클래스 평가를 위한 기준

2.3.3.1 객체와 클래스

객체와 클래스는 객체지향 방법론을 설명하는 대부분의 책장에서 혼용되고 있다. 클래스(class)라는 것은 클래스피케이션의 결과로, 인간이 실세계의 사물(things)을 생각하는 방식에 보다 가깝게 모형화한 구조체인 것이다. 분류화는 논리적 데이터 모델링의 추상화 매커니즘으로 일찍이 소개된 바 있었고, 최근에 객체지향 방법론의 중요한 부분으로 시스템 개발 분야에서 다시 주목을 받고 있다. 객체는 객체지향 시스템을 구성하는 기본 단위로서, 이런 객체 중에는 비슷한 속성을 갖는 객체들이 많이 있다. 이와 같이 비슷한 속성(공통된 연산, 다른 객체와의 공통된 관계 그리고 공통된 의미)을 가진 객체들의 집단을 클래스라 한다. 클래스는 일련의 개별객체(instances)집합(entities, object, things)에 대해 공통적으로 추상화하고 요약하는 역할을 수행한다. 비공식적으로 하나의 클래스는 인스턴스 집합에 의해 공유되는 속성(동작도 포함)에

대해 공통적으로 추상화하고 요약하는 역할을 수행한다. 비공식적으로 하나의 클래스는 인스턴스 집합에 의해 공유되는 속성(동작도 포함)의 집합으로, 계층을 통하여 보다 일반적인 클래스로부터 좀 더 구체화된 클래스가 속성을 계승받는 식으로 다른 클래스들과 관련을 가지기도 한다. 클래스는 공통된 속성과 행위를 갖는 객체들을 표현한 것으로, 클래스로부터의 인스턴스가 바로 객체인 것이다. 클래스는 객체 고유의 상태를 저장하는 메모리 영역인 인스턴스 변수와 이를 조작하는 연산의 정의들로 구성된다. 클래스로부터 객체로의 인스턴스화에 상대되는 개념으로 객체로부터 클래스로의 집단화를 추상화(abstraction)라 한다.

2.3.3. 클래스 평가의 기준

데이터 모델링은 어떤 실체에 대한 사용자의 지식 표현을 창조하는 것이라는 가정 하에 일관된 지식의 표현 도구로서의 클래스 집합의 질을 평가하기 위해 Parsons와 Wand는 제시한 네 가지 기준을 제시하였다. 그들은 클래스를 추상화된 개념(concepts)으로 파악하고 인지 심리학과 언어학에 기초한 원칙들을 제시했다.

① 인스턴스의 추상화원칙(Principle of Abstraction from Instances)

클래스는 인스턴스의 공집합이 아닌 집합에 의해 공유되는 속성들을 추상화 해야한다. 실세계를 인지할 때에 다른 인스턴스들과 공유하는 속성에 따라 인스턴스들을 분류하게 되는데, 하나의 클래스를 속성들로 구성된 집합으로 본다면 거기에는 반드시 모든 속성들을 다 보유하는 인스턴스의 공집합이 아닌 집합이 존재해야만 한다. 그렇지 않으면 그 클래스는 무의미하다.

② 최대 추상화의 원칙(Principle of Maximal Abstraction)

한 클래스의 전체 인스턴스에 속한 모든 속성은 그 클래스의 정의내에 포함되어야한다. 그래야만 그 클래스의 인스턴스들에 대한 정보 요구에 인스턴스화된 객체가 응할 수 있게 된다. 그러므로 클래스 정의에 전체 인스턴스들에 나타나는 속성 중 어떤 것을 빠뜨렸다면 인스턴스화된 객체는 그 속성과 관련된 정보 요구에 응하지 못하게 될 것이다.

③ 클래스 구조의 완전성 원칙(Principle of Completeness)

개발의 대상이 되는 문제 영역은 여러 인스턴스들의 집합, 그 인스턴스들이 지니는 모든 속성의 집합, 그리고 클래스들의 집합으로 구성된 지 ‘지식의 모집단’ (universe of knowledge)으로 볼 수 있다. 이때 모든 속성들은 최소한 하나의 클래스에는 정의되어 있어야 한다. 즉 어떤 속성도 지식의 모집단을 추상화함에 있어서 빠져서는 안된다는 것이다.

④ 클래스 구조의 비중복성 원칙(Principle of Non-redundancy)

하나 혹은 그 이상의 다른 클래스로부터 명세되는 클래스는 반드시 상위 클래스의 어떤 멤버 인스턴스에서도 발견되지 않는 정보를 보유해야만 한다.

이상의 네 가지 원칙들은 객체로부터 클래스를 유도하고, 그들간의 관계를 정립할 때에 동시에 고려되어야 할 것이다.

2.4 시험·분석·연구 장비현황 및 공동활용 관한 선행연구

2.4.1 정부 투자 시험·분석·연구 장비 현황

객체 일반적으로 ‘연구시설·장비’ (research facilities and equipments)는 연구개발(R&D) 수행을 위해 필요한 연구시설과 연구시설장비를 총칭하는 용어로 사용되고 있다(Irvin et al., 1997). 그러나 두 용어는 약간의 차이가 있는데, 우선 ‘연구시설’은 특수한 환경

을 조성하거나 특수지역으로 이동할 수 있는 설비를 갖춘 편의적이고 독립적인 연구공간으로서 무균실, 동물실, 수조, 풍동 등이 대표적이다.[5] 이에 비하여 ‘연구시설장비’는 100만원 이상의 구축비용이 소요되며 1년 이상의 내구성을 가진 연구개발을 위한 유형의 비소비적 자산으로서 질량분석기, 전자현미경등을 들 수 있다(국가과학기술위원회, 2010: 6-7). 이러한 차이에도 불구하고 실무적으로는 두 용어의 합성어인 연구시설장비가 좀더 많이 사용되고 있는 것이 현실이다.

그러나 이 연구에서는 분석대상인 기초연구사업에서는 연구시설장비가 전체 연구시설·장비의 97%를 차지하고 있어(이찬구 외, 2012) 좀더 한정적인 의미를 가지는 연구시설장비라는 용어를 사용하게 될 것이다. 한편, 연구시설장비는 실무적인 차원에서 용도별, 비용별, 활용목적별, 활용상태별 등으로 분류하여 각각의 관리방안이 다소간 상이하게 제시되고 있다. 그러나 여기서는 이 연구의 핵심 주제인 활용목적별 분류만을 정리하고자 하며 아래와 같이 나타낼 수 있다.

[표2-5] 연구시설장비의 활용목적별 분류

구 분	정의 및 관리방안
공동활용서비스 (public service) 장비	- 대외 개방을 위해 사용료, 이용절차, 이용시간, 전담인력 등 세부적인 운영규정에 의해 운영되는 시설 및 장비
공동활용허용 (joint use) 장비	- 연구자가 공동연구 등의 목적으로 타 연구자에게 활용을 허용한 장비 - 주로 장비보유 책임자의 판단에 의하여 허용 여부가 결정되며, 공동활용을 위한 세부적인 운영규정은 없는 상태
단독활용 (private use) 장비	- 단독활용을 목적으로 구축되는 장비 - 주로 개별 연구자가 직접 관리하며, 구입부서에서만 활용 가능

출처자료: 국가과학기술위원회(2010: 7-8)

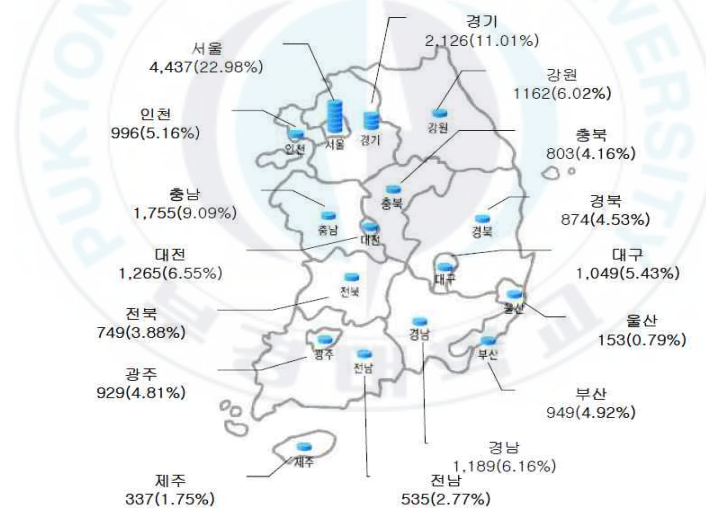
2003년 이후 정부의 시험·분석·연구 장비 구축 사업비는 전체적으로 증가하고 있으나 실제 가동율은 [그림2-7]과 같이 14% 수준으로 저조한 것으로 나타나고 있다.

특히, 국내 기관 중 기술표준원과 같이 기능변화 및 대내외 환경변화로 인하여 구축장비의 활용이 지속적으로 감소하는 기관들도 상당수에 이르는

것으로 나타나고 있다.

또한 산업자원부에서 지원하고 있는 인프라넷(온라인상에서 산업기술 인프라에 대한 정보공유와 활용을 목적으로 구축된 포털사이트, <http://www.infranet.or.kr>)에 등록된 시험·분석·연구 장비의 지역별 분포 상황은 서울, 인천, 경기도 전체의 약 38%, 대전, 충청이 전체의 약 20%로 지역 편중 현상도 심하다고 할 수 있다.

이러한 이유로 국내 기관이 보유하고 있는 시험·분석·연구 장비의 공동 활용이 이루어지면 활용율의 증가와 동일 장비의 중복투자가 방지될 뿐만 아니라 연구개발에 있어 국가 전체의 지역 균형 발전에도 긍정적인 역할을 할 수 있다고 볼 수 있다.



[그림2-7] 지역별 시험분석연구 장비의 분포 상황

2.4.2 시험·분석·연구 장비 공동 활용 선행연구 및 사례

‘연구장비공동활용 정책집행의 영향요인과 발전방향논고’ (황병상, 2006)에 의하면 연구장비 공동활용은 협동연구개발촉진법 등 법령상의 근거를 가지고 있으며, 1988년 기초과학분야 첨단기기 공동활용을 목적으로

한 한국기초과학지원연구원의 설립, 각 대학 공동실험실습관의 운영, 연구 장비 데이터베이스 제공 등에 의해 그 중요성은 더욱 증가할 것이라고 평가하고 있다. 연구장비 공동활용정책' 개념은 협의로는 '학연산의 연구장비를 공동으로 활용하여 예산투자의 효율성을 높이는 정책'을 의미하고, 광의로는 '연구장비를 공동으로 활용하여 예산투자의 효율성을 높이고, 학연산 연구협력과 공동연구라는 새로운 가치를 창출하는 정책'으로 보다 적극적인 의미까지 포함하는 개념으로 정의하고 있다. 이외에도 연구장비 공동활용 활성화 방안에 관한 연구로는 최덕인(1996), 구중익(2006) 및 박광순 외(2004)가 있으며, 공동 연구시설 및 장비 관련 연구조사는 과학기술혁신본부(2005), 권기현 외(2005), 권용수(2003) 등이 있다. 주요 기관의 선행 연구 사례를 분석해보면 다음의 [표 2-6]과 같다.

[표2-6] 시험·분석·연구 장비의 활용에 관한 국내 기관 주요 선행 연구

연도	연구내용	연구수행자
2003	· 대형 연구 시설 및 장비의 국제 공동 활용 방안	과학기술정책연구원
2006	· 지역 장비 공동활용 활성화 방안	한국기초과학지원연구원
2006	· 공동부문 R&D 장비의 효율적 활용 방안	한국기초과학지원연구원
2007	· 제2차 과학기술기본계획 수립을 위한 기획연구	한국과학기술기획평가원
2007	· 범부처 연구시설·장비 공동활용 촉진 세부추진방안	산업자원부 외 6개 부처
2007	· 시험연구장비 활용도 제고를 위한 '기술인프라 파트너십' 추진방안	산업자원부 기술표준원
2007	· 유관기관 시험연구장비의 효율적 활용방안	산업자원부 기술표준원

2.4.3 시험·분석·연구 장비의 공동 활용 파급효과 분석

2.4.3.1 파급효과 분석 대상 장비

본 논문에서는 산업자원부 산하 기술표준원의 “기술인프라 파트너십” 프로그램에 참여하고 있는 총 81개의 기관 중 공동 활용을 희망하는 69개

기관의 6,737종의 장비를 표본 집단으로 하였다. 분석 대상 장비 현황은 다음의 [표 2-7]과 같다.

[표2-7]분석 대상 장비 현황

구분	기관 수	장비 수(종)	장비 금액(억원)	장비 가동율(%)	연간 외부 시험검사 수수료(억원)
전체	81	8,227	4,298	16	9,790
표본 집단	69	6,737	3,960	14	4,744

장비 금액은 구입가를 기준으로 하였으며, 감가상각율과 물가 인상을 동일하게 보아현 시점의 가격과 구입가를 동일한 가치로 고려하였다. 장비의 평균 가동율은 기술표준원의 “기술인프라 파트너십” 프로그램 참여 기관을 상대로 한 설문 조사를 기반으로 하였다.

2.4.3.1 파급효과 분석 대상 장비 분석 효과

본 논문에서는 시험·분석·연구 장비의 공동 활용에 의한 파급효과를 상기 파급효과 분석 대상 장비에 대해서 경제적, 기술적 및 사회적 파급효과로 구분하여 분석한 뒤 국가 전체 장비의 관점에서 고찰하였다.

기술 인프라의 구축과 관리는 기술이 무기가 되는 21세기 글로벌 환경에서 매우 중요한 국가적 과제이며, 기술인프라의 중요한 요소 중 하나인 시험·분석·연구 장비의 전략적 운영은 이러한 맥락에서 중요한 의미를 갖는다. 시험·분석·연구 장비 전략적 운영 방법의 하나인 장비의 공동 활용은 활용률 증가로 인한 경제성의 제고로 시험·분석 당사자에게 이익을 제공할 수 있으며, 국가기술 발전의 계기를 제공하고, 중소기업을 진흥시킬수 있는 방안이 되는 등 기술적, 사회적으로도 다양한 파급효과를 가져올 수 있다.

[표2-8] 대상 장비를 공동 활용할 경우 가동율 증대에 의한 절감 효과

항목	수치	비고
(A) - 장비금액(억원)	3,960	분석 대상 장비 금액
(B) - 현재 가동율(%)	14	설문결과
(C) - 공동활용시 가동율(%)	20.2	설문결과
(D) - 연간 현재 시험·분석 수수료(억원)	4,744	설문결과
(E) - 현재 장비 금액 대비 연간 시험·분석 수수료 비율(%)	1.198	$(D) \div (A)$
(F) - 가동율 향상시 예상 연간 시험·분석 수수료(억원)	6,845	$(D) \times \{(C) \div (B)\}$
공동활용시 추가로 발생 될 수 있는 경제적 이익	2,101	$(F) - (D)$

장비의 공동 활용은 현재보다 장비의 가동율을 증가시켜 간접적으로 장비의 생산성을 향상시킬 수 있는 효과를 가져올 수 있는데 현재 대상 장비의 외부 공동 활용은 전무한 상황으로 설문조사 결과 외부 공동 활용을 현재 활용의 45%까지 증가시킬 수 있는 것으로 나타나고 있다. 이는 현재 가동율 14%를 공동 활용을 통하여 장비의 가동율이 20.2% 까지 증가할 수 있음을 의미하며 증가된 가동율에 비례하여 시험·검사 수수료로 인한 경제적 추가 이익이 발생할 수 있음을 의미한다.

추산 결과는 장비의 공동 활용으로 가동율이 증가될 경우, 시험·검사를 수행하는 기관입장에서는 증가된 가동율 만큼의 추가 수수료로 인한 이익이 발생할 수 있다는 의미이기도 하며, 시험·검사를 의뢰하는 입장에서는 그만큼의 비용을 절약할 수 있다는 의미이기도 하다. 이러한 상황이 표본집단이 아닌 국가 전체 장비로 확대될 수 있다면 공동활용으로 인한 국가 전체적 경제이익은 막대하다고 볼 수 있다. 더불어 장비의 공동 활용은 기관간의 동일 장비의 중복투자를 방지할 수 있어 정부의 예산 효율적 집행이라는 무형의 효과도 제공할 수 있다. 특히 산업자원부에 따르면 신규 장비 투자액이 170억원일 경우, 이의 5.8배에 해당하는 1,000억원의 연간 해외

시험·검사 비용을 절감할 수 있다는 해석에 비추어 보면, 장비의 공동 활용은 외화절약 효과도 가져올 수 있을 것으로 판단된다.



Ⅲ. 생산일정계획의 개념적 설계

3.1. 일정계획

3.1.1. 생산기능 및 형태의 기본개념

생산이란 새로운 것을 만들어 낸다는 의미로 제조업에서는 소재를 구입하여 물리적 또는 화학적인 특성을 변화시키는 에너지를 사용해서 소재를 제품으로 변환시키는 것을 의미하며(Hitomi Katsundo(2000)) 이러한 일련의 과정을 경제적이고 효율적으로 달성하기 위하여 계획 및 통제하는 일련의 활동을 생산관리라고 한다.[6]

생산은 크게 생산기능, 설계기능, 관리기능의 세가지 기능으로 나눌 수 있다.

첫째, 생산기능은 소재를 여러 가지로 변환시켜 제품으로 만드는 기능이다.

둘째, 설계기능은 제품설계, 공정설계로 나누며 생산을 위한 정보의 기술적인 사항을 다루는 기능이다.

셋째, 관리기능은 생산하는 양과 시기를 조절하는 기능으로 생산계획과 일정계획이 이에 해당한다.

생산의 기능은 제품의 특성, 고객의 제품 주문방식과 생산방식 등에 따라 여러형태로 분류된다.

<제품주문 방식에 의한 분류>

- 예측생산 : 수요를 예측하여 표준화된 제품을 먼저 생산하는 형태

- 주문생산 : 고객의 주문에 의하여 생산되는 형태(공정흐름 방식 분류)
- 연속생산 : 수요량이 많은 제품을 전용라인에서 연속 생산하는 형태
- 간헐생산 : 일정량을 모아서 생산 제조라인을 편성해 생산하는 형태
- 단속생산 : 제조설비를 개개 제품의 제조 과정 따라 설계 이용 형태

<제품 구성에 의한 분류>

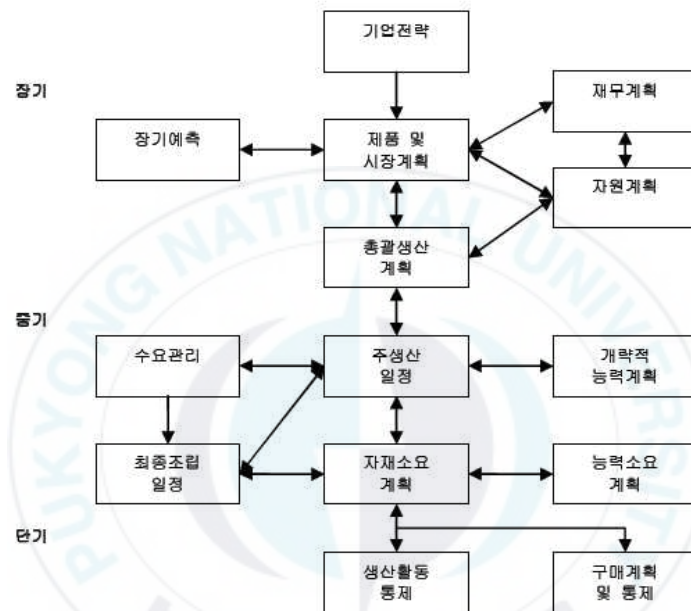
- 조립생산 : 부분품 또는 중간 조립품에 대하여 예측생산을 수행하고 고객 주문이 확정되면 조립을 시작하는 형태
 - 프로세스 생산 : 화학제품과 같이 원소를 결합하여 생산하는 형태
 - 분해 생산 : 원료가 제조 과정을 거치는 동안 복수의 제품이 되는 형태
- 제조 기업은 이와 같은 분류의 여러 가지의 조합으로 이루어진다. 이러한 조합의 방법에 따라 생산의 기능은 다양한 방법으로 나타나며, 생산성이나 효율을 높이기 위한 관리항목과 방법도 달라진다.

3.1.2. 생산계획의 형태

생산계획이란 제조업의 장기 또는 단기적인 목표를 달성하기 위한 정책으로서 정확한 계획이 수립될 수 있도록 여러 가지 요인들을 고려한 내외적인 정보를 수집, 파악하여야 한다. 제조업에서는 설비, 자재, 노동력 등 생산자원의 한계가 있기때문에, 최적의 계획을 수립하기 위하여 자원의 효율적인 배분과 활용은 필수적이며, 이러한 계획이 가능하면 정확히 실행될 수 있도록 기업의 여러 활동들의 지원과 함께 정확한 관리를 위한 노력이 필요하다. 계획은 일정기간 동안의 활동을 수치화하는 것으로 흔히 장기계획, 중기계획 및 단기계획으로 나누어진다.[7]

장기계획은 일반적으로 일년에 한번 수립되며, 일년 이상의 계획기간을 갖

는다. 중기계획은 6개월, 3개월 또는 월별로 계획이 수립된다. 단기계획은 하루에서 수주일을 대상으로 실제 현장에서 집행하기 위한 세부계획 및 통제에 관한 것이다.



[그림3-1] 일반적인 생산계획 활동 (황규승1998)

3.1.3. 일정계획

생산계획은 일정기간의 범위 내에서 제품별 생산량이나 생산금액을 예측하는 활동이나, 일정계획은 생산계획이나 고객의 주문에 대하여 구체적으로 어떠한 제품 또는 부분품을 어떤 설비 또는 제조공정에서 어느 정도의 제조자원을 이용하여 얼마만큼을 어떻게 생산하느냐 하는 좀 더 구체화된 계획이다. 즉 생산일정계획은 제조를 위한 일련의 작업을 어느 설비에서

언제 생산할 것인가를 효율성 측면에서 결정하는 것이다.

일반적으로 Job Shop에서의 일정계획의 목표는 주 생산일정 또는 자재 소요계획을 일별, 시간별로 세분화하는 것이다. 다시 말해서 생산시스템에 대한 단기적 작업 부하량을 정확하게 명시하는데 있다. 또한 주 생산일정을 달성하기 위하여 작업의 진행과정을 점검하고, 필요하다면 주문 독촉이나 생산능력 조정을 하는 현장통제가 그 목적이다.

주문생산에서의 Job Shop 형태 생산에서는 고객의 주문을 받은 후에야 설계 작업으로 옮겨지기 때문에, 설계업무는 뒤에 이루어지고, 또 생산계획이 중요하게 다루어지지 않는 대신 납품 기일에 맞추도록 일정계획을 중요시한다.

주문생산에서의 일정계획을 작성하기 위한 업무의 흐름은 다음과 같다. 계약에 의해 고객의 주문이 생산계획 담당자에게 도착하기 위해서는 설계 담당자의 검토를 거쳐야 한다. 이때 설계 담당자는 비표준 제품에 대해서는 설계 일정계획을 수립한다. 그리고 생산계획 담당자는 설계일정 계획에서의 출하 예정 일자를 기준으로 개략적 생산능력계획(RCCP)을 세운다. 이 개략적 생산능력계획을 근거로 자재의 소요량과 소요일자를 산출하는 개략적 조달계획을 세운다. 소요자재의 조달리드타임이 길거나 긴급 자재의 경우, 이때 계획된 개략적 조달계획을 근거로 발주가 이루어진다.

설계 담당자에게 제품에 대한 도면이 완성되어 출하되면 생산관리 담당자에게도면이 배포가 된다. 생산계획 담당자는 이때의 정확한 출고일자를 기준으로 최종일정계획을 다시 세우게 된다. 이렇게 세워진 일정계획을 근거로 정상적인 자재투입 및 조달계획을 세우며, 일자에 맞게 작업지시를 내리게 된다.

이와 같은 절차에 의해 진행되는 주문생산에서의 Job Shop 일정계획은 세부 일정계획을 작성하기 위해 같은 기간 내 Shop에 도착한 작업들과의

관계를 통해 공정과 작업장 별로 구체화 된다.

일정계획 대상작업 오더의 종류는 다음과 같다.

1. 계획 오더

중기 또는 단기 생산계획에 의해 세워진 계획을 구체화 또는 예측하기 위한 가상의 작업

2. 전적 오더

고객의 주문이 발생한 가능성이 있는 작업으로 고객에 대한 납기일을 예상하기 위한 가상의 작업

3. 계약 오더

고객과의 주문이 체결되어 현장에서 작업이 이루어지는 실제의 작업 일정 계획 대상작업의 상태는 다음과 같다.

1. 대기작업

개략적 제조일정계획을 통해 세워진 작업

2. 확정작업

설계부의 도면 접수 후 부하 및 조달계획에 의해 구체적으로 세워진 작업

3. 진행작업

생산현장에서 진행중인 작업

4. 완료작업

일정계획에 의해 세워진 확정작업이 진행 작업을 거쳐 처리 완료된 작업

다음은 세부일정계획의 요소로써 세부일정계획 문제는 주어진 상황에 따라 그 복잡성과 난이도가 상이하다.

상황을 결정하는 요소로는 다음의 다섯 가지를 들 수 있다.

1) 작업도착 유형

2) 생산현장 내의 기계의 숫자와 다양성

- 3) 생산현장을 통과하는 작업의 흐름 유형
- 4) 작업들을 기계에 할당하는 우선순위 규칙
- 5) 세부일정계획의 평가기준

이와 같은 작업의 종류와 상태의 분류 및 계획 작성상의 요인을 통해 일정 계획시물레이션이 해당 목적에 맞는 일정계획으로 구분되어 처리된다.

이를 통해 예측과 실행과의 오차를 줄일 수 있어 신뢰할 수 있는 일정계획 자료가될 수 있다.

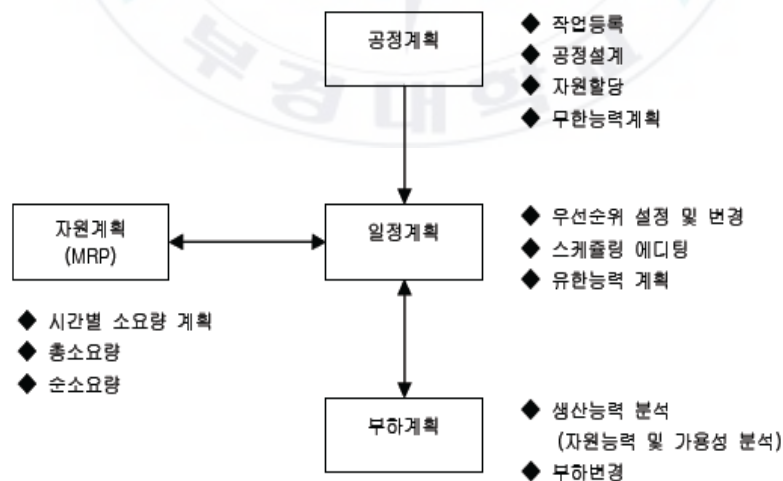
일정계획은 크게 세 가지의 기능으로 나뉘서 진행된다.

첫째, 공정계획

둘째, 일정계획

셋째, 부하계획

이상의 세 가지 기능에 대해서는 다음 항에서 자세히 다루고자 한다.



[그림3-2] 일반적인 일정계획 구성도

3.1.4 일정계획기능

3.1.4.1. 공정계획

결정된 예정계획에서 제품을 제조하기 위한 구체적 제조과정의 순서 작업 경로, 제조 방법들을 결정하는 공정관리 기능의 하나이다. 생산활동의 시간적 경과는 항상 제조과정의 순서나 작업경로에 의해 구현되기 때문에 생산계획의 구체적 계획으로서 제조에 대한 시간적 결정을 주목적으로 하는 실행계획의 중심적 위치를 차지한다. 공정계획은 제품의 흐름에 따라 수행되는 작업 및 그 계열에 대한 계획이기 때문에, 작업계획과 작업계열계획으로 구분할 수 있으며, 이것은 제품 구성품의 다소에 따라 차이가 생긴다.

하나의 주문이 발생하면 영업 부서에서는 계약번호와 제품마다의 계약번호에 대한 순번을 부여하여 내부적인 결재과정을 거쳐 설계팀과 생산 관리팀으로 보내진다. 이 때 제품에 대한 사양서와 고객과 약속한 납기일에 대한 정보를 제공한다.

설계팀을 거쳐 생산 관리팀에 도착한 Job은 도면 출고일과 납기일을 기준으로 개략적인 생산일정계획을 세우게 되는데 해당 Job에 대한 작업공정을 설계하고 필요한 공수와 순서를 계획하며 Job에 대한 무한 능력계획을 수립한다.

3.1.4.2. 일정계획

수립된 Job마다의 공정계획을 기준으로 공장에 대한 전체 일정계획을 수립한다.

즉 각 공정의 작업장마다 작업을 수행할 수 있는 능력이 상이하다. 일정계획을 통해 공정의 작업이 가능한 공정의 작업장을 선택해 주고 그 작업장

의 능력을 파악하여 부하 평준화를 통해 일정계획을 수립해 간다.

이때 일정계획을 수립하기 위한 Job 마다의 순서가 필요하다.

본 시스템에서는 Job 마다의 작업순서를 계획자가 임의적으로 정의하게 하고 이를 시스템이 각 작업장마다의 능력을 파악하여 부하 평준화를 통해 일정계획을 수립한다. 이러한 상세일정계획인 유한능력계획을 시스템을 통하여 수행하여 각 작업장마다의 작업순서와 부하계획을 수립한다.[8]

3.1.4.3. 부하계획

유한능력계획에 의해 수립된 부하평준화 결과를 바탕으로 각 Job 마다의 납기일을 평가하고, 납기지연이나 조기생산이 발생하면 Job에 대한 우선순위 변경뿐만 아니라 각 작업장마다의 능력을 재조정하여 적시납기가 가능하도록 일정계획을 재수립한다.[9]

작업장의 능력은 각 제품이 완성될 때까지의 일련의 작업을 수행하는 기계 설비와 작업원의 작업시간을 사전에 행한 시간연구에 의하여 어느 정도 정확하게 산출이 가능하다.

부하계획의 능력 재조정에는 기계의 회전을 증가나 작업원들의 작업강도를 높이는 강도적용, 일자별 작업시간이나 작업인원을 늘리는 양적 적응 또는 해당공정을외부에서 처리하는 외주적용 등이 있다.

계획자는 Job의 우선순위와 능력 재조정을 이용해 유한능력계획을 재수립한다.

3.2 자재소요 계획(MRP)과 도면 발행

이 시점에서 자재소요계획에 대한 논의는 생산 일정계획이 그 주변의 다른 활동들(예: 자원계획, 수요관리, 자재계획, 용량계획 등)과 필연적으로 연계된다고 했을 때, MRP는 생산일정계획과 자재의 재고 통제에 다 관여하는 체계라는 점에서 의미를 갖게 된다. MRP는 정확한 일정 체계, 효율적인 자재 통제 체계, 변화가 생기면 계획을 변경하기 위한 재일정 기구를 제공한다. 소요 자재가 필요할 때 제공될 수 있도록 하면서 재고 수준을 최소로 유지하는 것이 목적이다.[10]

또한, 최종품이나 구성품에 대한 도면도 넓은 의미에서 일종의 자재로 볼 수 있다. 그러나 최근 컴퓨터 이용한 설계와 같은 개념의 등장은 도면 정보에 대한 별도의 논의를 요구하게 된다.

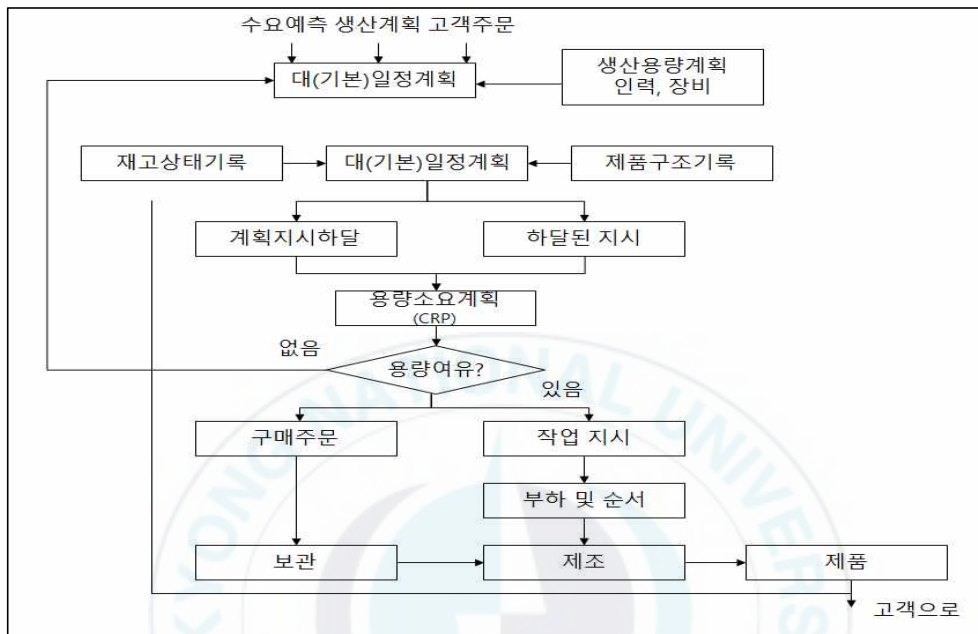
3.2.1 MRP의 목표와 특색

MRP 체계는 다음의 세 가지를 동시에 달성하는 것을 목적으로 한다.

첫 번째는 계획 생산과 고객 배달을 위한 자재, 구성품 및 제품의 기득성을 확실히 하고, 두 번째는 가능한 한 최소의 재고 수준을 유지하고 마지막은 제조활동, 배달 일정, 구매 활동이 원활히 되도록 계획하는 것이다.

한 품목에 대한 수요를 생각할 때, 그 품목에 대한 수요가 최종 품목이나 여타 품목에 독립적일 수도 있고, 그 품목에 대한 수요가 상위 품목에 대한 수요의 결과이거나 직접 연관되어 종속적일 수도 있다. 이 때 최종 제품에 대한 수요가 독립적이라 하더라도 제품을 구성하는 하위, 부속품목에 대한 수요는 조석적인 경향이 있게 된다. 종속 수요 품목은 자재설명서

(BOM:bill of material)를 전개하여 계산하게 되는데, MRP는 이러한 중
속 수요 품목을 대일정(master schedule)으로부터 계산한다.



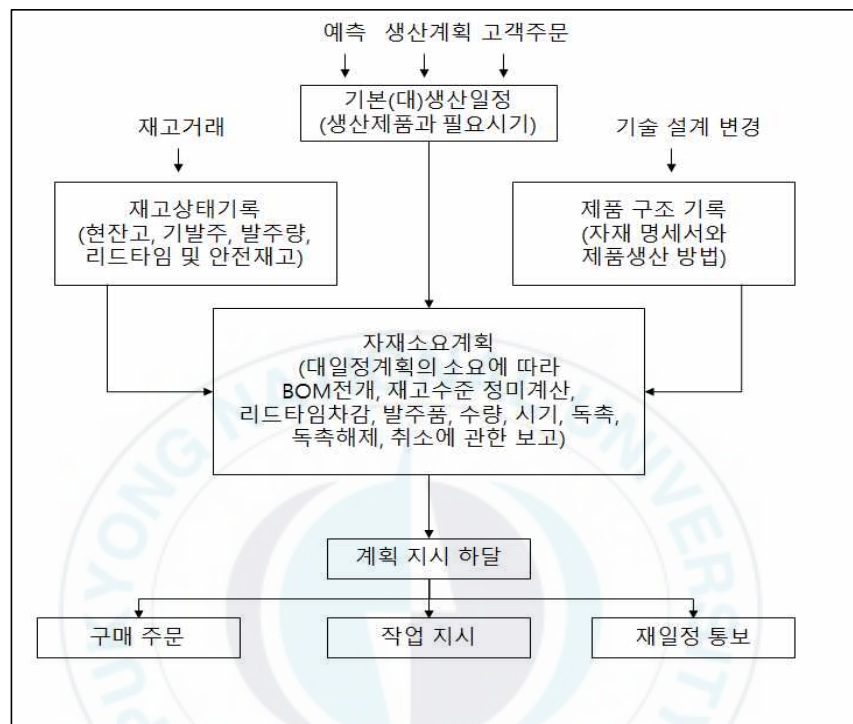
[그림3-3] MRP체계

3.2.2 MRP의 입출력

MRP체계의 3가지 주요 입력은 대일정계획, 재고상태기록, 제품구조기록이다. 대일정계획에는 모든 최종 품목에 대한 생산 계획의 윤곽이, 제품 구조 기록에는 각 최종 품목에 필요한 모든 자재, 구성품, 조립부품에 대한 정보가, 재고 상태 기록에는 재고 품목의 현보유 및 기발주 상태가 포함되어 있다.[11]

이상의 입력 정보를 토대로 MRP는 최종 품목을 생산하기 위해서 각 기간에 소요될 품목 수량에 대한 구매와 작업 일정에 대한 계획 지시를 출력하게 된다. 계획 지시 하달은 수량과 작업 지시를 공장에 하달하는 혹은

구매 주문을 납품자에게 통보하는 기간을 제공 한다. 아래의 그림은 MRP 체계의 입출력을 도식화 한 것이다.

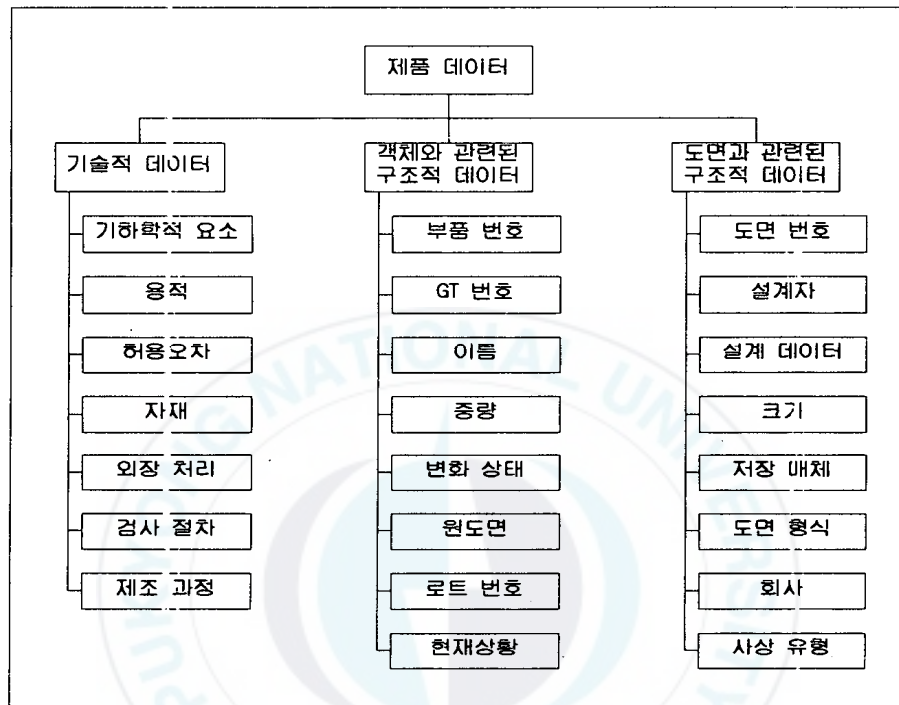


[그림3-4] MRP입출력

3.2.3 도면발행 계획

제작할 제품에 대한 도면을 준비하는 문제는 대부분의 생산관리 논의에 있어서 그에 대한 일정보다는 제품 품질을 결정짓는 품질 관리의 측면에서 접근하고 있다. 그러나 주문생산 내지는 프로젝트 생산시스템의 경우 도면의 확보(주문자가 도면을 제공하든, 도면을 새로 준비하든)는 BOM의 작성을 위한 기초자료가 된다는 의미에서 도면의 지연은 자재 조달의 지연을, 자재 조달의 지연은 생산일정을 지연을 유발하게 되므로 하나의 이러한 계획으로 관리 할 필요성이 높다[12]. 특히나 컴퓨터를 이용한 설계기

법의 등장은 데이터베이스라는 개념을 차출하게 되었는데, 그 개념은 단순한 설계도면의 정보가 아니라 제품 그 자체와 생산계획에 관련된 데이터까



[그림3-5] 제품과 생산계획 데이터를 포함하는 CAD DB

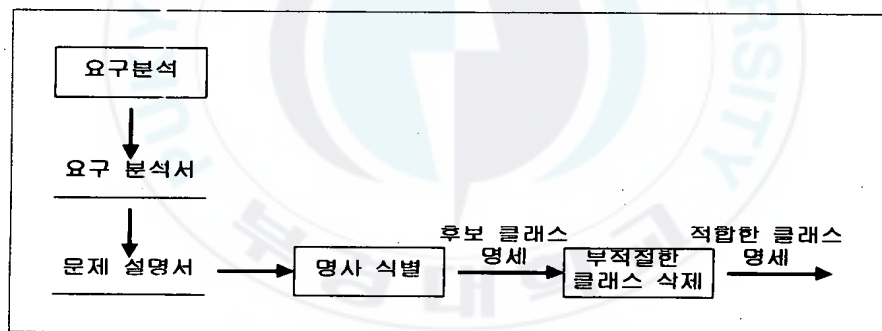
출처: U. Rembold, B. O. Nnaji & A. Storr, *Computer Integrated Manufacturing and Engineering*, Addison-Wesley, 1993, pp. 296. 에서 번역.

지 포함하는 것으로 인식되기도 한다. 그림[3-5]의 모델을 보면, 제품에 대한 기술적 내용뿐 아니라 구조적 내용까지도 포함하고 있음을 알 수 있다. 전통적으로 도면을 발행하는 과정은 필요를 인식하고, 초기 도면을 설계하여 합의를 도출하고, 분석하고 최적화한 후, 평가하고 발행하는 단계를 거치게 된다. 도면의 발행 일정을 관리한다는 입장에서는 도면 발행 지시와 최종 발행까지의 재활동들을 일종의 리드타임으로 생각하여 접근할 수 있을 것이다[13].

3.3 객체지향 데이터 모델의 설계

3.3.1 문제 설명서 작성과 클래스의 식별

클래스의 식별은 대개 문제 설명서에서 명사를 찾아 그 유효성을 평가함으로써 이루어진다. 객체 및 클래스는 시스템 상에서 처리해야 할 정보를 수집하고자 하는 기본적인 관심 대상으로 명백하게 구분되어질 수 있는 IT는 사물이다. 식별 방법은 [그림3-6]의 절차에 따라 사용자, 관리자, 개발자의 요구분석 결과인 요구분석서를 참조하여 문제 설명서를 작성하고, 그로부터 명사를 식별하여 후보 클래스를 찾아내고 다시 부적절한 클래스를 삭제한 후 적합한 클래스 명세를 작성한다.[14]



[그림3-6] 객체 클래스 식별 순서

출처: 최영근 허계범 공저, 객체지향 소프트웨어 공학, pp. 195.

생산일정계획은 부품가공의 착수에서 조립작업의 완료까지에 이르는 제품별, 부품별, 공정별, 그리고 직장별 작업의 일정과 생산에 직접적으로 관계를 가지는 설계, 구매, 외주, 검사 등의 업무에 대한 일정을 지시하게 된다. 일정계획의 계획수립 방향에서 보면 제품단위의 종방향 계획과 직장단위의 횡방향 계획으로 구분된다. 종방향 일정계획에 국한하여 생각할 때,

일정계획은 각 제품에 할당되어야 하는 적절한 재고수준 및 노동력을 결정할 뿐 아니라 생산구조를 통하여 흐르는 원재료나 구성요서에 대한 적절한 비율 및 시기를 결정해야한다. 구체적인 생산계획은 개별 제품을 기본단위로 하고, 원료 부품 혹은 조립품이 최종품목의 제조에 필요하다.

또한, 판매 예측이나 고객 주문으로부터 정보는 일정계획의 출발점이 된다. 생산계획은 수요예측을 토대로 생산하여야 할 제품의 품종과 수량, 생산공정, 생산일정, 작업인원배치, 설비배치 등에 관하여 가장 경제적이고 합리적인 방법에 관한 예정적인 의사결정의 체계이다. 시장생산방식을 채용하는 공장에서는 판매계획 도는 예측을 바탕으로 하여 생산량이 결정되고, 고객으로부터 주문을 받아 생산할 경우는 소정의 납기대로 제품을 완성하기 위해서 각 작업의 착수 및 완성의 기일을 결정한다.

대일정계획의 세우기 전에 장기생산능력계획에서부터 생산능력규모가 확정되어야한다. 생산일정계획은 최종 품목의 예측과 고객 주문으로부터 개발된다. 이는 가용한 생산 용량을 융통할수 있도록 정상화된 현실적 생산계획을 투영해야한다. 부품 및 부조립 부품들을 내부에서 생산하는 경우에는 그것을 생산하기 위한 기계 및 작업장별로 다양한 용량이 필요한 것이다.

생산 일정계획의 시행을 보장하려면 기업은 생산공정상 필요한 원료, 부품, 조립품 등이 적시에 공급될 수 있도록 계획을 세워야만 한다. 자체소요 계획이란 부품 및 원료 등을 생산일정에서 요구하는 대에 공급할 수 있도록 생산 혹은 주문하는데 대한 일정을 수립하는 방법이다. 즉 MRP는 생산일정계획과 재고 통제에 다 관여하게 되는데, 최종품목으로부터 모든 하위요소(조립품, 조립부품, 구성품)에 대한 필요한 일정을 제공한다. 작업장에서는 생산일정계획과 조화를 이룰 수 있게 하기 위한 경제적인 양의 자재확보가 필수적이다. 제품이나 부품의 도면도 넓은 의미에서는 일종의

자재가 된다. 따라서 도면 발행의 과정도 부품이나 구성품의 일정계획과 유사한 형태의 일정계획에 의해 관리되어야 할 것이다.

[표3-1] 생산 일정계획의 문제 설명서

생산능력과 고객의 주문에 대한 정보가 주어진 상황에서, 최종 제품은 여러개의 구성품을 조립함으로써 생산된다. 하나의 구성품 역시도 독립성을 지닌 몇 개의 부품을 조립함으로써 생산되고, 각 부품들은 원자재를 가공하여 준비하게 된다. 생산 활동의 각 단계별로 소용되는 자재와 도면은 해당 생산활동이 계획대로 이루어지는데 차질이 없이 조달될 수 있도록 관리한다.

문헌에 나타난 이상의 표현들에 입각하여 일반적인 생산계획용 객체 지향 데이터 모델의 문제 설명서를 제시하면 [표3-4]와 같다. [표3-4]의 문제 설명서로부터 얻은 명사는 생산능력, 예측 수요, 고객주문, 최종제품, 구성품, 부품, 원자재, 생산활동, 도면 등이다. 이 중에서 예측수요와 고객주문은 동일한 클래스의 다른 표현일 뿐이므로 고객 주문이라는 클래스명으로 통합할 수 있겠고, 클래스 평가 기준에 의해 생산 활동이라는 클래스는 부적절하다. 따라서 식별된 클래스는 최종제품, 고객주문, 생산능력, 부품, 원자재, 구성품, 도면 등이다. 여기에 각 하위활동의 구체적인 일정계획의 수립을 위한 계산절차들을 포함하는 클래스로서 스케줄러라는 객체를 추가할 수 있다. 이는 객체 모델링 과정에서 각 객체의 오퍼레이션으로 객체 내에 정의될 수 있다. 이는 객체 모델링 과정에서 각 객체의 오퍼레이션으로 객체 내에 정의될 수 있다. 그러나 이와 같이 별도의 클래스로 나타냄으로써 구체적인 속성들은 추가하지 않은 상태에서의 논의의 편의성을 높일 수 있고, 또한 객체지향 시스템의 다형성에 대한 지원을 염두에 두어서 실제로 계산 모듈만을 포함하는 하나의 클래스를 생성하여 여타의 클래스들이 스케줄러 클래스의 오퍼레이션을 계승하도록 구성할 수도 있을 것이다.

3.3.2 데이터 사전과 클래스 관계도

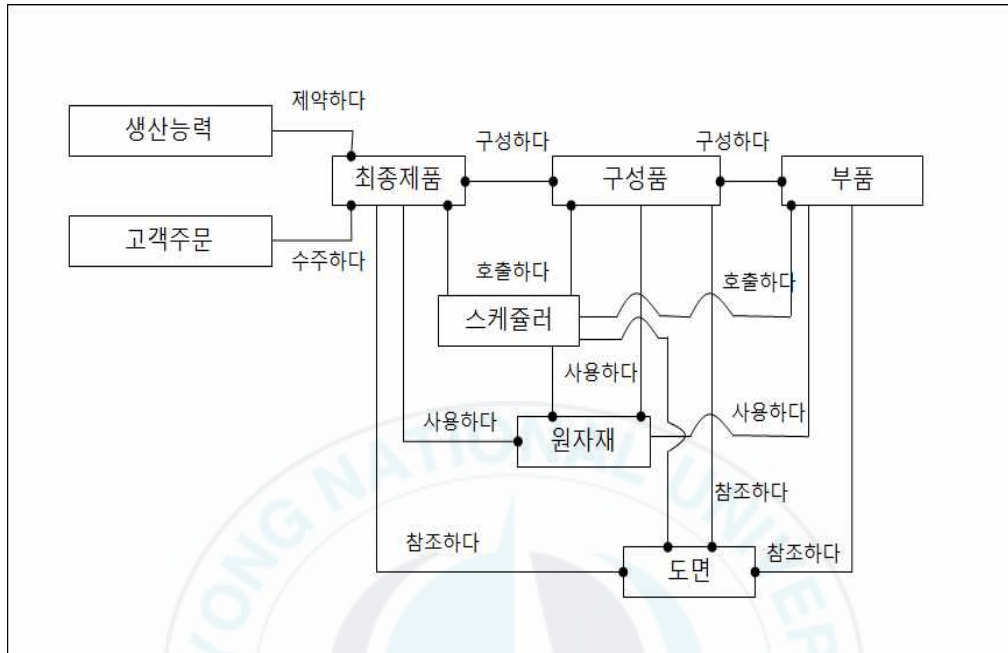
식별된 클래스의 명세는 표[3-2]와 같다.

[표3-2] 일정계획 클래스의 데이터사전

클래스명	설명
최종제품	생산시스템이 생산하게 될 최종품
생산능력	생산능력계획에 의해 확정된 생산용량
원자재	부품이나 구성품이 아닌 최하위 수준의 자재
구성품	최종제품의 하위시스템을 구성하는 독립성을 가지는 제조품
도면	각 제조품의 설계도면
고객주문	수요 예측이나 고객 주문에 의해 생산해야 할 제품의 수량
부품	원자재를 가공하여 자체 생산한 1차 가공품
스케줄러	입력자료를 토대로 구체적인 일정을 계산하는 모듈

객체지향 접근법에서는 분석 및 설계의 대상이 되는 현실세계(문제영역)의 모든 현상을 객체와 객체간의 관계로 모형화하게 된다. 앞장에서 식별된 객체 클래스들은 [그림3-7]과 같은 관계와 계층을 가지고 있다[15]. 이러한 클래스간의 관계 정립은 문제설명서의 동사부분에 주목함으로써 이루어질 수 있는데 예를 들면, 최종 제품은 여러 개의 구성품을 조립한다는 문제 설명서의 표현으로부터 최종제품이라는 클래스가 구성품 클래스와 모종의 관계를 가지고 있음이 클래스 관계도에서 나타나게 될 것이다. 한 가지 생각할 것은 최종 제품을 구성하는 구성품에 대한 것으로 최종 제품이 하나의 프로젝트 정도의 성격을 같은 경우나 그 규모가 클 경우에는 구성품이 다시 하위의 구성 품목들로 나누어 식별될 수 있다는 것이다.

3.3.3 클래스의 속성 추가

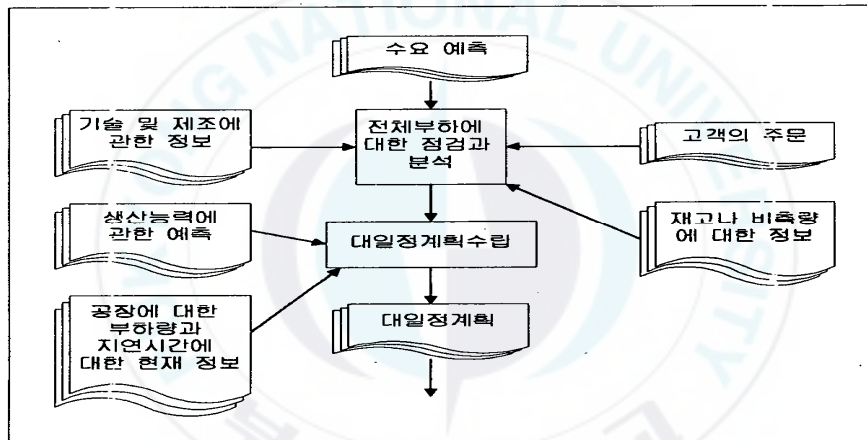


[그림3-7] 생산일정계획의 클래스 관계도

클래스에 속성을 추가하는 것은 복합클래스를 기본클래스와 여타 복합클래스에 관련시키는 활동이라 할 수 있다. 기본클래스는 그 자료형이 기본적인 것이라는 것을 의미한다. 즉 문자열이나 정수형, 날짜형 등과 같이 대부분의 프로그래밍 언어들이나 데이터베이스 시스템이 지원하는 자료형들을 하나의 클래스로 사용할 수 있다. 여기에 기본클래스를 이미 보유하는 복합클래스를 자신의 자료형으로 갖는다거나 일련의 연산과정도 그 내부에 정의될 수 있다는 것이 객체지향 방법론의 특성인 것이다. 따라서 클래스의 속성을 정의하는 것은 데이터베이스의 정보요구를 확인하는 절차도 포함하게 된다. 문헌들에 나타난 일정계획 관련 클래스의 정보 요구에 입각하여 각 클래스 속성들을 추가하기로 하겠다.

3.3.3.1 대·소 일정계획의 정보요구

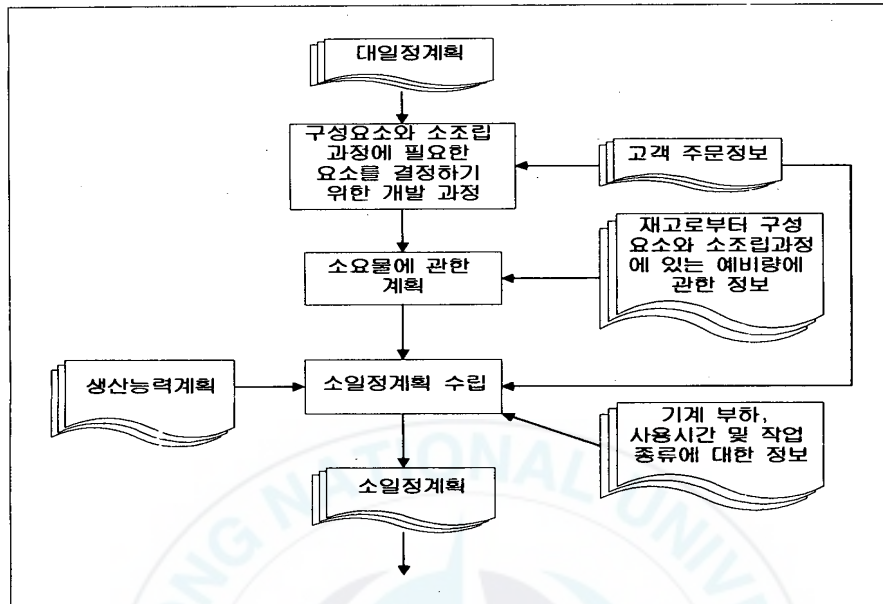
장기적인 생산능력규모가 주어지면, 그에 따른 대(기본)일정계획을 수립하게 되는데, 이때 요구되는 정보는 고객의 주문량, 재고량에 대한 저보 등이 있다. 그러한 정보흐름도는 [그림3-8]과 같다[16]. 수요예측과 실제 고객의 주문량을 배합하는 방식은 고객의 주문이 있기 전에 하는 계획, 실제 주문이 있어야만 하는 계획, 양자의 종합형태의 계획 등 업체에 따라 다르며, 대일정계획이란 최종 산출물 생산에 관한 개략적인 시간 계획이 된다.



[그림3-8] 대일정계획에 관한 정보흐름도

출처: 강석호, 산업공학개론, 영지문화사, 1994, pp. 317.

대일정계획이 확정되면 세부적인 생산량과 생산시기, 기계부하 등을 소일정계획에서 수립하게 되는데, 소일정계획은 실질적인 공장 내의 작업반장에 의해 실행되는 일정계획으로 [그림3-9]와 같은 정보가 계획을 수립할 때 사용된다. [그림3-8]과 [그림3-9]가 대일정계획과 소일정계획으로 이어지는 생산 일정계획의 종적인 측면과 그 정보의 흐름을 나타낸다면, 이미 소개된 [그림3-3]은 횡적인 측면과 정보흐름을 나타내고 있는 것이다.



[그림3-9] 소일정계획에 관한 정보흐름도

출처: 강석호, 산업공학개론, 영지문화사, 1994, pp. 320.

3.3.3.2 클래스 속성의 추가

일정계획에 관련된 정보 요구는 식별된 객체의 속성으로써 나타나게 된다. 최종제품 객체는 제품명에 의해 식별되고, 최종제품 조립을 위한 작업의 착수 및 완성시기(제작기간)와 일정의 정적인 시간 개념을 나타내는 공수를 속성으로 가진다. 또한 소정의 납기에 소정의 수량을 생산하기 위하여 각종 작업의 예정이나 관련업무의 수배 시기를 결정해야 하므로 제작기간을 날짜형으로 계산하고 생산능력의 가용공수를 참조하여 실제 투입될 수 있는 공수를 계산하는 오퍼레이션이 추가된다.

최종제품 조립에 소요되는 구성품들도 구성품명에 의해 식별될 뿐 아니라, 최종제품처럼 독립성을 가진 하나의 조립품이므로 생산하는데 필요한

공수와 제작기간을 속성으로 가지게 된다. 제품이 대형 선박과 같이 큰 규모일 경우에는 구성품들이 다시 하위 구성품으로, 하위 구성품은 또 그 하위 구성품으로 나누어져서 관리될 수 있다. 최종제품과는 달리 구성품의 생산은 여러 작업장에서 동시에 이루어지게 되므로 각 작업의 평균대기시간을 이용하여 어느 작업장에서 수행할 것인지를 파악하기 위해 제작부서에 대한 정보도 포함하여야 한다. 제작기간과 공수를 계산하는 오퍼레이션은 최종 제품 객체에서와 같은 기능을 하게 된다.

상세한 일정은 납기, 수량, 기준일정 및 생산능력을 고려하여 작성하게 되는데, 그 일정은 제품단위 혹은 부품단위로 작성하게 된다. 부품명을 통해 부품이 식별되고, 모든 부품을 내부 생산할 경우 그 부품의 생산을 위한 공수, 평균 제작기간, 그리고 제작부서에 대한 정보를 속성을 갖게 된다.

구성품과 부품 객체의 경우 생산 작업이 동시에 이루어지게 되므로, 각 개 활동의 시간과 선후관계가 추정되어야 한다. 이는 프로젝트를 관리에서 구분된 활동들의 순서, 선후행관계, 그리고 기대소요시간을 기초로 주공정을 구하는 것처럼 구성품과 부품의 생산활동을 관리할 수 있게 된다. 이를 위해 연결구성품과 연결부품이라는 속성이 추가된다.

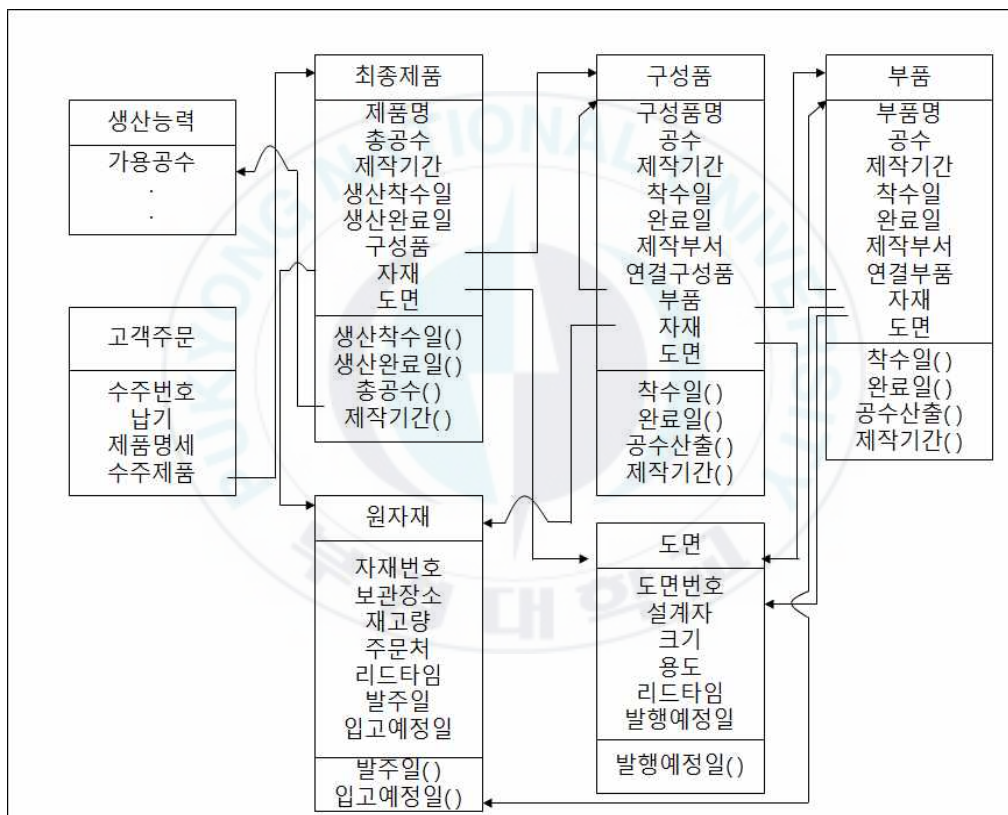
원자재 객체의 경우는 자재번호에 의해 식별된다. 원자재의 수요는 종속적인 경향이 있으므로 그 자재가 어느 창고에 있는지(보관장소), 재고량은 얼마인지, 어디서 주문할 수 있으며 각 주문처의 인도기간(lead time)은 얼마인지에 대한 정보는 물론 최종 품목 완성일로부터 역산한 발주일자와 발주량에 대한 정보를 속성으로 가져야 하며 그것을 계산하는 오퍼레이션도 포함되어야 한다.

도면 객체는 도면번호에 의해서 식별되고, 설계자, 크기, 용도, 발행절차에 소요되는 시간 등을 속성으로 가지며 원자재와 마찬가지로 최종제품의

완성일자로부터 역산되는 발행예정일도 포함된다.

고객의 주문에는 설계(제품명세), 주문량, 납기 등이 포함된다. 기업체는 수주된 제품에 수주번호를 부여하여 식별하게 된다.

생산능력 객체에 있어서는 생산능력계획을 통한 공장 및 설비구입은 장기적으로 준비되므로 통제 가능한 것은 투입될 수 있는 인력을 표시하는 가용공수 정도이므로, 그 정보를 속성으로 갖는다.



[그림3-10] 속성을 포함한 클래스 관계도

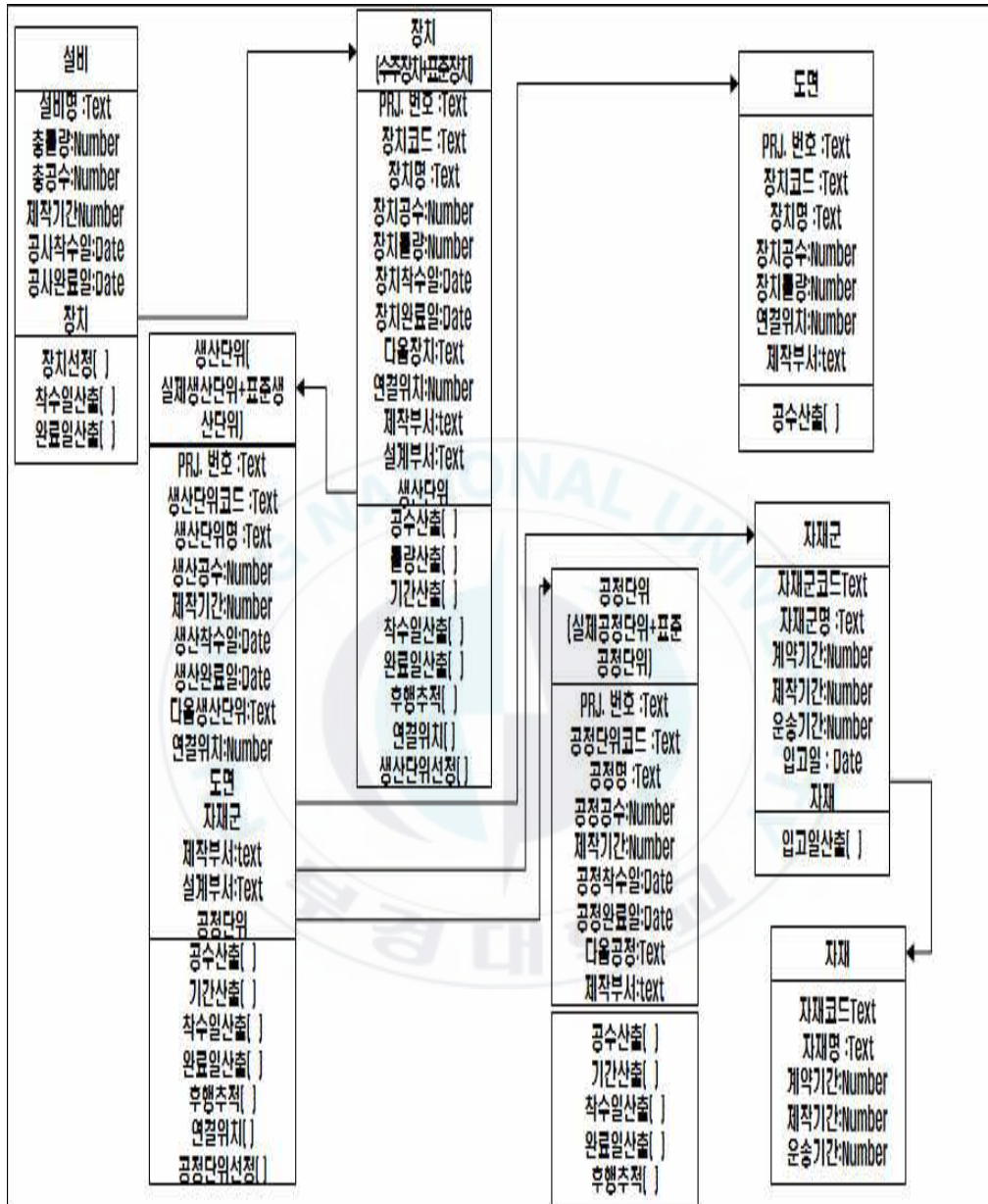
객체 클래스는 그 정의에 따라 데이터와 오퍼레이션을 캡슐화한 자기충족성을 가진 실체로 나타나야 한다. 지금까지 논의된 일정계획 절차나 정보요구를 이제 각 클래스의 속성과 오퍼레이션으로 캡슐화 함으로써 [그림 3-7]에 속성을 추가한 클래스 관계도를 [그림3-10]에 도시하였다. 각

클래스들은 기본적으로 그 클래스가 언제 시작해서 언제 끝마치는가에 대한 정보와 공정순서, 그리고 공정의 소요시간에 대한 데이터를 속성으로 가져야하며, 그것을 기반으로 각 클래스의 일정이 계산되는 오퍼레이션과 필요한 경우 사용자의 개입을 허용하는 오퍼레이션 등을 보유해야만 한다. 이 관계도에 있어서 각 클래스가 해당 속성의 값을 사용자로부터 입력받고 수정하고 저장하고 다시 불러내고 하는 일반적인 오퍼레이션들은 따로 표기하지 않았다. 이러한 기본적인 오퍼레이션은 모든 데이터베이스 관련 응용 프로그램들이 예외 없이 지원하는 것들이므로 별도의 오퍼레이션 정의가 필요치 않을 것이다.

3.3.4 객체지향 데이터 모델의 도출

객체지향 데이터 모델은 이미 언급한 대로 객체 모델과 원칙적으로 대동소이한 것이다. 그러나 데이터베이스 스키마로의 전용을 위해 더 효과적인 방식으로 표기할 수 있을 것이다. 특히 여러 클래스들의 내포관계 즉, 내포된 클래스같은 경우는 단순히 두 클래스가 연관을 가진다라는 식의 관계 표현보다는 구체적으로 각 속성 하나 하나에 대한 자료형을 화살표 등을 이용하여 나타내는 방식이 추천될 만 하다. 객체지향 데이터 모델의 일반적인 도식법에 대한 합의는 아직 이루어져 있지 않지만 클래스 속성(중첩 객체의 형태를 취함)은 화살표를 이용해서 분명히 해주는 식의 도식법이 일반적이다. 도출된 객체지향 데이터 모델은 [그림3-11]과 같다.

데이터와 데이터를 조작하기 위한 오퍼레이션을 객체 내에 캡슐화 한다는 객체지향 시스템의 특징을 이용하여 스케줄러에 의해 구체적인 일정을 산출하던 것을 각 클래스에 오퍼레이션을 추가하여 표현하였다.



[그림3-11] 객체지향 데이터 모델 도출

IV. 대형설비 생산 일정계획 데이터베이스 개념설계

4.1 데이터모델 설계의 개요

4.1.1 대형설비 생산의 특성

설비라는 것은 대량으로 생산되는 제품과는 달리 주문에 의해 소량생산되는 것으로 특정 목적에 맞는 생산이나 공정을 처리하는 기능을 가진 장치 또는 시설을 말하며, 대형설비라는 것은 대규모 설비의 집합체로 예를 들면 화공설비, 발전설비, 하역설비, 운반설비 등과 같은 것을 말한다. 설비에 따라서 그 제작기간이 1,2년 이상의 장기간에 걸쳐 전체 설비 제작에 대한 계획을 세우고 일정을 결정하고, 그 진행을 통제한다거나 하나의 설비를 여러 개의 작업들로 또 그 작업들은 더 세분화된 하위 작업들로 나누어 관리해야 한다는 측면에서 프로젝트와 유사하다[17].

실제로 나타난 프로젝트 관리과정을 한 설비에 적용하여 생각해보자. 예를 들어 특정한 자재를 취급하기 위한 설비를 생산한다고 가정할 경우 그와 유사한 설비만을 전담하여 생산하는 기존의 한 사업부를 통하여 총물량 100톤짜리 설비를 생산하겠다는 계획을 가지고, 전체설비를 부리는 기계장치와 크레인장치와 옮겨 싣는 기계장치 등의 과업으로 분류하여 그 각 과업으로 분류하여 그 각 과업들을 계속 작은 단위로 분류하여 한 작업자에게 배당될 수준으로 세분화한 후, 각 단위들에 필요한 공수와 자재 그리고 그들간의 선후관계를 정립한다면 그것이 바로 일정계획에 해당할 것이며, 그 계획대로 생산활동을 통제한다 했을 때, 그것은 다른 아닌 하나의 프로젝트를 관리하는 과정이 되는 것이다.

그런데 대형설비를 발주한 발주자의 요구사항의 정형성 정도에 따른 것으로 발주자들의 요구사항이 매우 정형화되어 있는 경우라면 그 설비의 경우 어느 정도의 표준이 존재할 수 있다는 것이므로 표준사항설비로 구분할 수 있다는 것이다. 또한 같은 논리로 표준사항 혹은 완전한 주문사항설비로 구분할 수 있다는 것이다. 준표준 내지는 표준사항설비에 주목하여 본다면 설비의 기본형이 어느 정도 정해져 있으며, 발주자의 요구에 따라서 규모와 용량이 정해지게 될 것이다.

이상의 논의를 토대로 대형설비 생산의 특성을 열거해 보면 먼저 주문에 의해서 제작되고, 생산기간이 길며, 제작을 위한 도면이 사전에 준비되지 못하는 경우가 많다, 따라서 자제소요내역이 사전에 제공되지 못하므로 유사공사가 존재하기 한자로 요약 될 수 있다. 따라서 동일한 설비에 대해서 산 사업부가 전담하여 계속적으로 생산 활동을 수행함으로써 마치 기성품을 제작하듯이 용량별로 설비를 어느 정도 규격화 할 수 있게 된 경우에 대해서는 단속적 생산시스템의 성격을 보유하는 프로젝트 생산시스템으로 정의할 수가 있을 것이다. 본 연구에서는 이와 같이 단속적 생산시스템의 성격을 보유하는 대형설비 생산시스템을 중심으로 객체지향 데이터 모델을 도출하게 될 것이다.

이러한 대형설비 생산시스템을 관리함에 있어서 가장 중요한 과제는 일반 제품과 마찬가지로 품질과 납기이다. 보통 발주자는 장기전략과 계획에 의하여 공장 또는 대형설비를 설치하게 되는데, 최단공기에 맞추어 발주하는 경향이 있으며 납기 지연은 허용되지 않는다. 생산자 자신의 기술 축적과 엔지니어링의 고도화가 수익성의 향상과 직결되기 때문에 품질에 대한 관리는 경쟁력을 제고를 위한 중요한 문제가 된다. 대형설비의 납기준수와 품질 유지 문제는 자재공급, 도면공급 및 생산일정계획의 일관성에 따라 결정, 이러한 요인들에 대한 효과적인 관리가 수익성에 영향을 주게 된다.

4.1.2 적용기업의 분석

4.1.2.1 적용기업의 개요

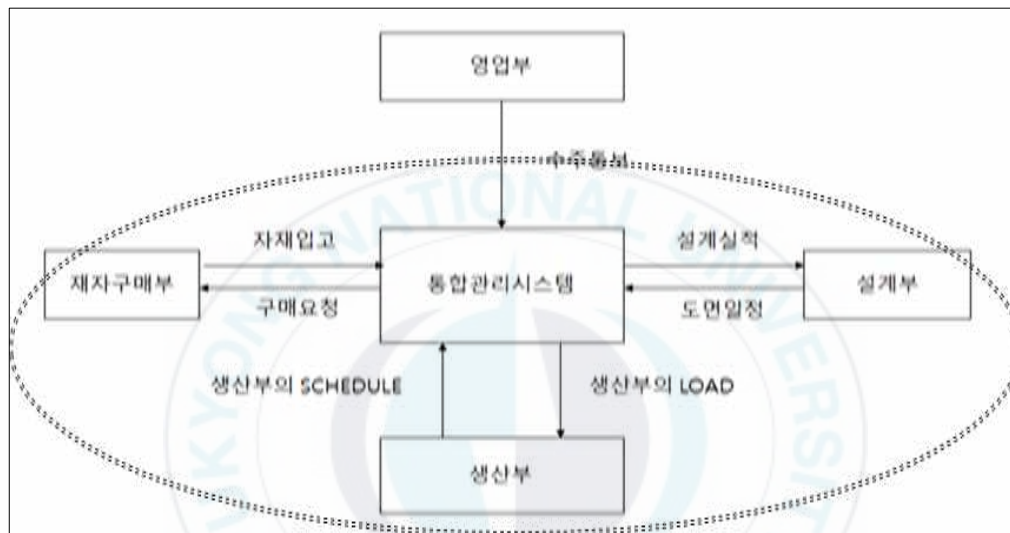
본 연구소는 발전소, 화학공장, 제철공장과 같은 열교환기 및 플랜트를 주문받아 생산하는 기업으로 수주에서 납품까지의 일관적 정보 관리 시스템과 효율적 정보시스템 및 환경변화에 대한 유연한 대응체계를 구축하기 위해 영업, 생산, 설계, 자재관리를 통합하는 관리 시스템을 개발하는 중에 있다. 이 시스템은 생산부서의 설비 및 장치관련 데이터, 자재부서의 자재관련 데이터, 그리고 설계부서의 도면관련 데이터를 기반으로 하게 된다. 이들은 모두 관계형 데이터베이스로 개발되며, 영업부서에서 수주가 발행할 때 데이터베이스로부터 통합일정 계획의 수립을 위한 기초자료가 추출되는 메커니즘으로 움직인다. 이러한 통합관리시스템의 세부과제는 생산계획의 유연성확보, 자재구매의 효과적 관리 생산일적의 정확화, 공수 산정의 객관성 확보 등이다.

4.1.2.2 시스템 개발의 도메인

[그림4-1]에 나타난 레벨0 DFD는 사례연구소가 목표로 하는 통합관리 시스템과 관련 부서의 역할을 잘 보여주고 있다. 사례기업은 자재구매부, 설계부, 생산부, 영업부의 각 기능부문별로 이미 과거 실적에 기초한 데이터가 어느 정도 확보되어 있는 상황이다.

통합 관리 시스템은 특정 설비에 대한 수주가 발행했을 때, 자재구매로부터의 자재정보와 설계부로부터 도면정보 그리고 생산부로부터의 유효공수에 대한 정보를 바탕으로 구매 요청과 도면 발행 일정, 그리고 대일정계

획, 중일정계획, 소일정계획으로 이어지는 생산일정계획을 제시하는 시스템이 된다. 단, 영업관련 부문은 일단 본 연구의 영역에서 제외하고 그림 [4-1]의 점선으로부터 표시된 영역만을 대상으로 한다. 문제가 되는 것은 주문생산 설비의 특성상 각 설비에 대한 사양을 발주자가 결정하므로 소요도면과 소요자재에 대한 완전한 표준을 사전에 준비 할 수 없다는 것이다.



[그림4-1] Level 초기정보흐름도

[그림4-1] 레벨0 데이터 흐름도의 특정 설비가 수주된 후에야 수주설비에 대한 도면이 작성되고, 그 도면을 근거로 소요자재가 결정될 수밖에 없다. 그렇지만 통합관리의 측면에서는 관리의 대상이 될 기초자료가 요구되므로, 과거의 실적을 토대로 기준이 될 만한 데이터를 중심으로 수주공사의 사양에 따라 필요한 정보를 어느 정도 산출해 낼 수 있는 표준 데이터베이스가 필요한 것이다. 특히 표준 데이터베이스는 공수, 물량, 기간 등의 값을 실제 수치 대신 가중치로 나타냄으로써 표준을 구성하게 된다. 이러한 표준 데이터베이스는 실제 공사 수주시에 기초자료가 되며, 수주가 이루어진 경우 입력되는 자료와 통합하여 통합관리시스템의 요구에 부응하게

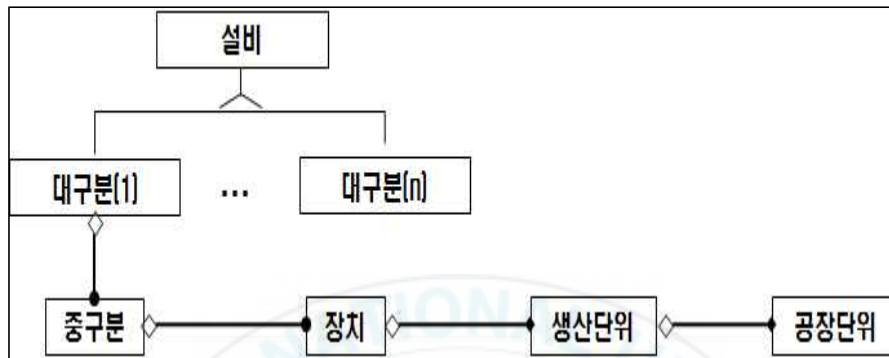
된다. 방대한 표준 데이터베이스로부터 수주된 하나의 설비 혹은 장치에 대한 정보만을 추출하여 적절한 조작을 통해 실제치로 변환하고, 그 정보를 유지하는 과정을 모형화 하기에는 관계형 데이터 모델보다는 객체지향 데이터 모델이 적합하다. 문제 영역을 객체라는 매개체를 통하여 접근하는 객체지향 방법론은 이런 경우에 있어 만족할 만한 해결책을 제시한다.

통합관리시스템은 하나의 독립된 실체로서의 연구시설설비를 그 관리 대상으로 한다. 즉 한 설비에 대해서 그 설비는 어떤 하위 장치들로 구성되며, 그 각각의 장치와 관련된 도면과 소요자재의 조달도 통합 관리하여 세부적인 생산일정계획의 수립으로까지 연계하려는 것이 그 목적이다. 그러므로 시스템의 개발은 각 설비를 관리가능단위로 분할하는 것으로부터 출발한다. 설비를 관리가능단위로 분할하는 것은 설비를 구성하는 각 장치에 대해 실체성을 부여함은 물론 분해의 정도에 따라 해당 단위를 대·중·소 일정계획에 대응시키게 된다는 점에서 의미를 가진다.

OMT로 표현한 한 설비에 대한 클래스 계층도의 모습은 그림[4-2]와 같다. 하나의 설비를 여러 개의 장치들로 구성되고, 그 장치들은 다시 더 작은 장치들의 집합으로 구성된다. 이에 착안하여 우선 설비마다 고유의 분류코드를 부여한 후 각 설비를 유형별로 크게 대분류하고, 이를 다시 중분류, 소분류 하였다. 장치별로 소분류된 모든 단위들은 다섯 자리의 코드를 가지게 된다. 이렇게 소분류된 장치 단위는 구체적인 도면정보와 자재정보에 유기적으로 연결 될 수 있는 생산단위로 다시 나누어지게 되는데, 이 생산단위는 자재 부서에서 구매하여 조달할 자재와 설계부서에서 발행하게 될 발행하게 될 도면의 기준 단위가 된다. 또 생산단위는 생산 현장에서의 공정관리가 가능할 활동수준으로서의 공정단위로 한 번 더 나누어지게 되며 이때 생산단위와 공정단위는 별도의 코드를 각 각 부여받게 된다.

[그림4-2]에 나타난 클래스 계층도의 의미론적 특성을 보면, 기본적으로

하나의 설비는 해당 설비의 여러 유형들이 일반화된 형태를 가진다. 그러나 유형별로 대구분된 이후로는 해당 설비의 장치나 생산단위 또는 작업공정을 기준으로 세분화가 이루어지는 이는 집단화 관계로 표현될 수 있다.



[그림 4-2] 설비의 클래스 계층도

4.1.3 데이터 모델 설계의 방향

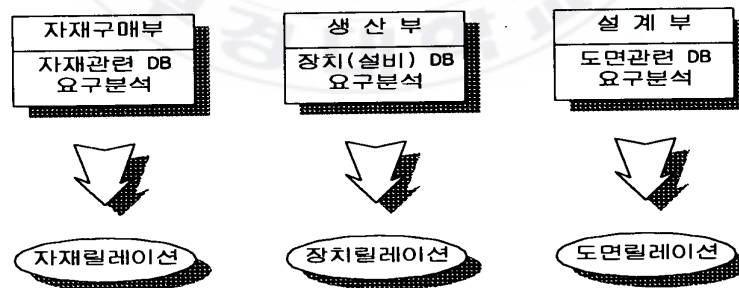
표준 데이터베이스는 객체로서의 ‘설비’를 데이터베이스화하는 것이다. 설비는 설비의 상태를 나타내는 여러 속성과 설비의 동작을 나타내는 오퍼레이션으로 구성된다.

관계형 데이터 모델을 통해서 사례 기업의 데이터 요구 명세를 표현하기에는 관계형 데이터 모델 자체의 한계점이 그대로 노출된다. 표현하고자 하는 개체인 ‘설비’는 그 구조가 복잡하고 특히 중첩 객체를 포함하고 있으며, 단 하나의 릴레이션에 용해시켜 버리기에는 무리가 되는 의미적 측면들이 있다. 또한 표준 자료로부터 수주시 재구성되는 실제 자료로의 변환을 위해서는 약간의 오퍼레이션이 요구되는데, 관계형 모델에서는 이러한 오퍼레이션에 대해서는 논의 자체가 불가능하다.

객체지향 데이터 모델의 설계는 관계형 데이터 모델의 확장 시도와 일맥상통한다. 즉 사용자 정의 자료형을 지원할 수 있도록 모델을 확장함으로

써 중첩객체가 표현될 수 있도록 하고, 데이터와 데이터 값에 대한 오퍼레이션을 한데 묶어 캡슐화하며, 기 개발된 코드체계에 기반 하여 표현하고자하는 개체들간의 의미적 관계가 명확히 표현될 수 있게 하는 것이다.

적용 기관의 경우, 시스템 분석 및 설계의 기법은 전통적 접근법에 의한 이른바 기능적 분할로부터 출발하였다. 전체 시스템을 생산부서, 설계부서, 자재부서 등의 기능 단위를 중심으로 분할하여 각 각에 대해 개발팀을 구성함으로써 분할된 하위 시스템에 대한 분석과 설계가 이루어졌다. 통합관리시스템은 그 결과로 구축된 데이터베이스로부터 필요한 자료를 제공받게 된다. 데이터베이스 스키마도 각 기능별 요구분석에 의해서 구성될 것이다. [그림4-3]에서 확인할 수 있는 것은 개체 설비는 자신과 관련된 속성들을 여러 개의 릴레이션에 분산하여 위치시키고 있다는 것이다. 실세계에서 확인된 개체를 시스템 내의 개체로 사상하는 과정에서 그 실체성을 가늠한 잃지 않게 하는 것이 시스템 분석설계의 한 가지 목표임을 생각할 때 관계형 데이터 모델은 전반적인 표현력의 부족은 그러한 배려를 충실히 해 내기에는 부족하다.

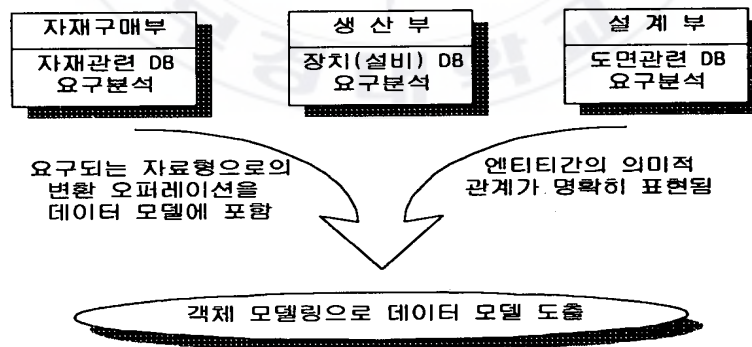


[그림4-3] 기능적 분해를 통한 데이터 모델링

당 사례의 경우에는 객체를 식별하여 그들간의 관계를 정립함으로써 시스템을 분석하고 설계하는 객체지향 방법론을 통하여 표현하고자 하는 엔터

티에 주목하여 데이터 모델을 생성하고자 한다. 객체지향 접근법은 [그림 4-4]에 나타난 바와 같이 하나의 설비에 주목하여 모델링하게 되는데, 설비의 속성은 분류코드와 설비명, 불량, 공수, 제작기간, 설계부서, 생산부서 납기, 도면, 자재 등을 포함하며 설비의 오퍼레이션은 설비명 변경, 불량가중치변경, 제작기간 가중치 조정 등과 같은 데이터 조작을 위한 것과 불량결정, 장치선택 등과 같은 클래스 고유의 오퍼레이션으로 나누어진다.

실제 데이터 모델링의 과정에 있어서는 상용 데이터베이스 관리시스템이 제공하는 레코드 생성, 수정 검색 등의 오퍼레이션들은 따로 고려할 필요가 없다. 그것은 관계형 데이터 모델도 기본적인 오퍼레이션들을 묵시적으로 포함하고 있기 때문이다. 따라서 본 연구의 데이터 모델은 사례에서 요구되는 특별한 오퍼레이션들만 다루게 된다. 그 오퍼레이션들에 대한 고려는 표준 데이터베이스로부터 실제 수주 발생시 이루어져야 할 데이터 변환 과정들에 대한 것들로서 객체지향 방법론을 통해 데이터 모델링을 시도하게 된 하나의 요인이 된다.



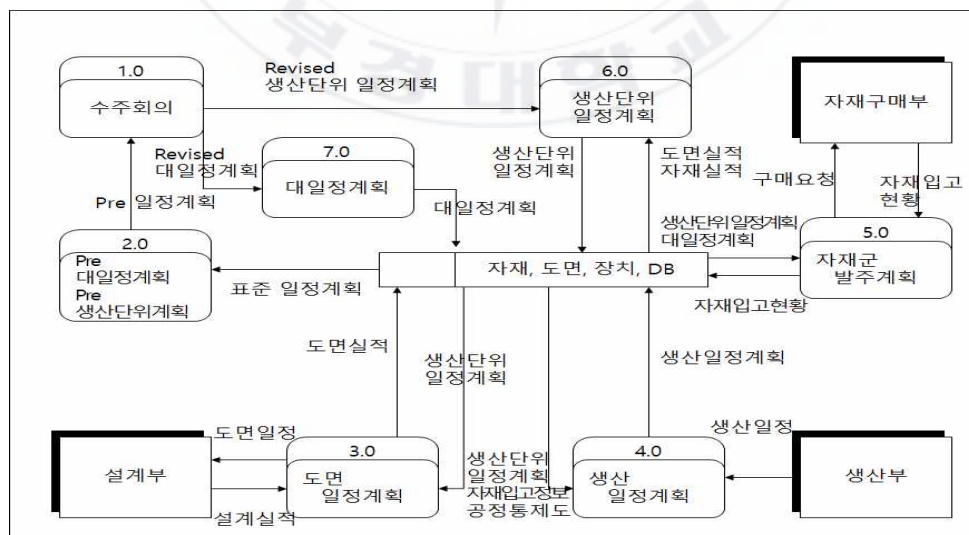
[그림 4-4] 객체지향 설계의 활용

4.2 객체 모델링과 데이터 모델의 도출

4.2.1 클래스의 식별

본 논문의 객체 모델링 과정에서 시종일관하게 나타나는 개념은 바로 복합 개체이다. 사실상 현실 세계의 엔터티는 단 하나의 기본 클래스만으로 구성되는 경우가 거의 없을 것이다. 기본 클래스를 포함하여 여러 클래스들을 구성원으로 하는 복합 객체의 형태이기 쉽다. 당 사례의 경우도 모든 클래스들이 복합 객체로 나타난다[18].

객체를 식별 하는하는 문제설명의 작성이 핵심이다. 문제 설명서는 요구 분석의 과정을 통해서 도출하게 되는데, 당 사례의 경우 이미 기능적 분할을 통해 각 기능 부문별로 요구 분석이 이루어진 상태이다. 따라서 객체 모델링을 위한 별도의 요구 분석은 필요치 않다. [그림2-4]의 시스템 개발 과정 도면에 나타난 바와 같이 비객체지향 접근법의 분석 단계에서 객체지향분석설계로의 전환은 약간의 변칙을 가질 수도 있다.



[그림4-5] Leve2 데이터 흐름도

그러나 비객체지향 접근법으로 전체 시스템을 개발한다 하더라도 최소한 한번 이상의 변칙이 있을 수 있고, 객체지향시스템으로의 개발이 가져다 줄 소프트웨어 공학상의 장점들을 생각할 때 객체지향 분석설계로의 전환은 추천될 만하다 하겠다. 통합관리 시스템의 레벨 1 DFD는 [그림4-5]와 같다. 객체 식별을 위한 문제 설명서 도출을 위해 사용될 수 있다.

요구분석서나 문제 설명서는 동일한 문제 영역에 대한 분석의 결과라는 측면에서 매우 유사한 개념이다. 그러나 기본적으로 전통적 접근법의 요구분석서가 시스템을 처리과정을 중심으로 바라본 결과물이라면, 객체지향 접근법의 문제 설명서는 객체를 중심으로 바라본 결과물이라 할 수 있다. 당 사례에서는 생산 부서에서 생산하게 될 산출물인 설비 혹은 설비를 구성하는 장치가 실체성을 가진 독립된 객체(속성과 오퍼레이션을 가진)라는 데 착안하여 작성한 문제 설명서는 [표4-1]과 같다.

[표4-1]문제설명서

하나의 설비는 타입별로 대구부되고, 대구분된 설비는 시스템적 유사기능을 가진 장치의 그룹별로 중구분된다. 중구분된 설비는 다시 독립성을 지닌 여러 개의 표준장치로 소분구분되고, 하나의 장치는 여러 개의 표준생산단위로 구성된다. 표준생산단위는 표준공정단위에 의해 고정별로 구분된다.

설비의 수주는 주문자의 요구에 따라 표준장치들 중에서 선별적으로 이루어지게 된다. 어떤 장치를 수주 설비에 포함시킬 것인가 하는 것은 설비 수주시의 명세에 의해서 파악되어 수주장치를 구성하게 된다. 수주장치 역시도 표준 생산단위로부터 실제 생산단위가 선별되고, 실제생산단위의 경우도 표준 공정단위로부터 선정된 시제공전단위만으로 구성된다.

도면과 자재의 관리는 생산단위를 기준으로 이루어진다. 한 생산단위에 소요되는 도면과 자재에 대해 각 각 도면계획과 자재군계획이 수립되고, 스케줄러에 의해 생산일정계획이 수립된다.

[표4-1]에서 확인된 객체는 설비 대구분된 설비, 중구분된 설비, 표준장치, 표준생산단위, 표준공정단위, 수주장치, 실제생산단위, 실제공정단위, 도면, 자재, 도면계획 자재군 계획, 스케줄러 등이다. 이들 모두 후보 클래스가 된다. 이들 중에서 대구분된 설비, 중구분된 설비는 앞의 클래스 평가 4원칙에 의해 부적절한 클래스로 판명되어 제거된다. 따라서 확인된 객체 클래스는 [표4-2]와 같다.

[표4-2] 확인된 객체클래스

설비, 표준장치, 표준생산단위, 표준공정단위, 수주장치, 실제생산단위, 실제공정단위, 스케줄러, 도면, 도면계획, 자재군계획, 표준자재

4.2.2 데이터 사전과 클래스관계도

[표4-3] 데이터 사전

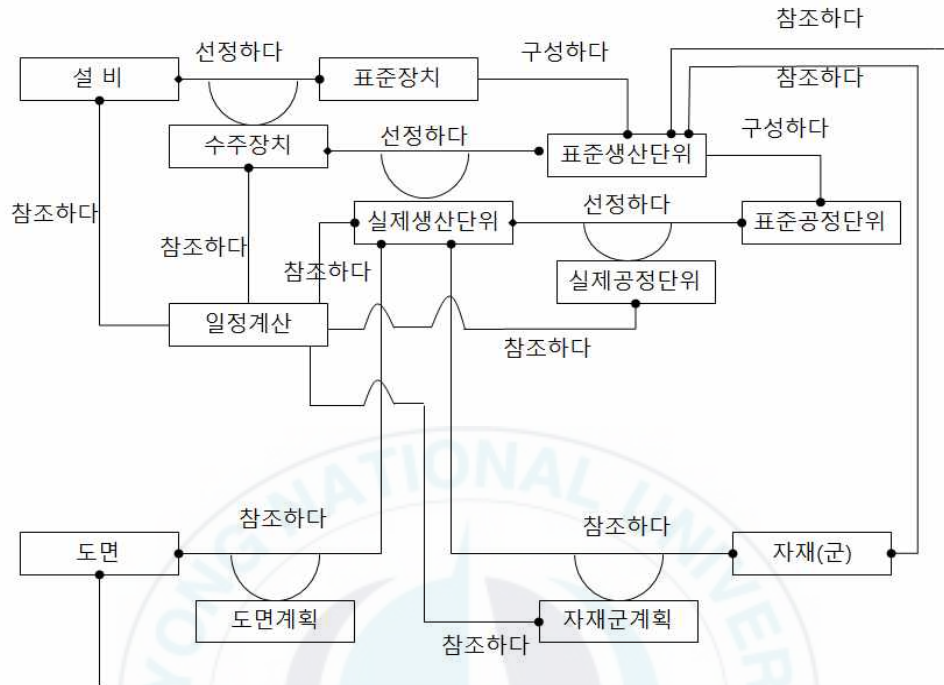
클래스 이름	역할
설비	여러 장치들로 구성된 수주단위로 사례기업은 유사한 설비들을 한데 묶어 사업부를 구성하고 있다.
표준장치	한 설비를 구성하는 표준이 되는 장치
표준생산단위	한 장치를 구성하는 표준이 되는 생산단위
실제공정단위	한 생산단위를 구성하는 표준이 되는 공정단위
수주장치	공사의 수주 과정에서 실제로 수주된 장치로 표준장치들에서 선정된다.
실제생산단위	수주 결과에 따라 생산하게 되는 생산단위로 표준생산단위 중에서 선정된다.
실제공정단위	수주 결과에 따라 수행되는 공정으로 표준공정단위 중에서 선정된다.

자재군	도면 제작 이전에 자재 조달 계획용으로 준비된 자재의 군에 대한 의사 BOM(Bill of Material)
자재	자재구매에서 발주하는 모든 자재에 대한 정보
도면	설계부에서 제작되는 모든 도면에 대한 정보
자재군계획	생산단위별로 요구되는 도면에 대한 일정 계획
도면계획	생산단위별로 요구되는 도면에 대한 일정 계획
일정계산	수주된 설비, 수주장치, 실제생산단위, 실제공정단위, 도면계획, 자재군계획 등의 일정계획을 산출하는 모듈

식별된 객체에 대한 설명은 데이터 사전을 통해 이루어진다. 확인된 설비, 표준장치, 표준생산단위, 표준공정단위, 수주장치, 시제생산단위, 실제공정단위, 스케줄러, 도면, 도면계획, 자재군 계획, 표준자재 등의 객체들에 대한 명세는 [표4-3]과 같다. 객체들간의 관계는 문제 설명서의 술어 부분을 통해서 확인할 수 있다. 이를 통해서 연관, 집단화 등의 의미들이 표현된다. 객체들간의 관계는 [그림4-6]과 같다.

4.2.3 클래스 속성의 확인

객체의 속성과 연결 속성을 확인하는 것은 그림[4-6]의 각 객체를 복합 객체로 보고 구성 객체를 찾아내는 과정이 된다. 사실상 대부분의 객체 모델링을 소개한 문헌에서는 객체의 속성을 확인하여 객체의 내부를 구성하는 과정에 대한 명확한 언급이 발견되고 있지는 않다. 기본 클래스나 복합



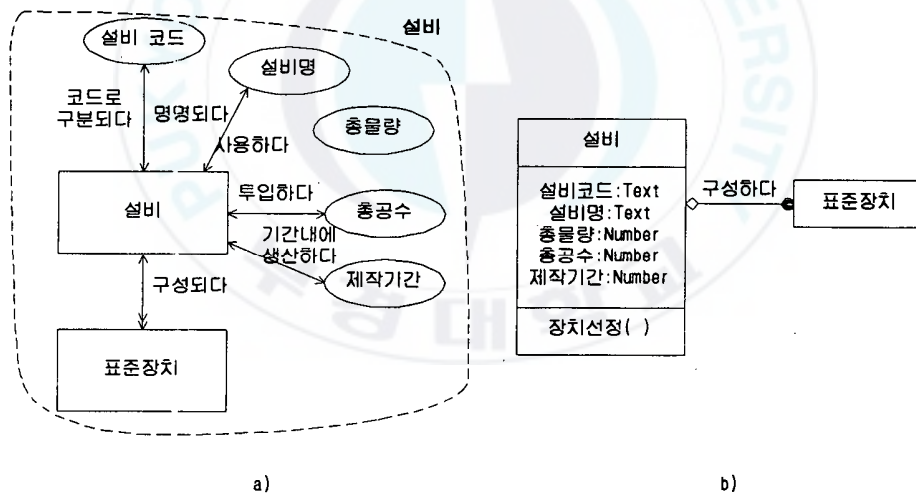
[그림4-6] 객체 클래스 관계도

객체의 개념은 그러한 부분에 대한 좋은 해답이 된다. 즉[그림4-6]에 소개된 객체는 엔터티 객체에 해당한다는 것이다. 그리고 이제부터 확인하게 될 각 객체의 속성과 오퍼레이션들은 속성 객체에 해당하고, 클래스로 표현된 연관(수주장치, 실제생산단위, 실제공정단위, 도면계획, 자재군계획) 관계는 관계 객체에 해당한다. 또한 각 관계 객체의 속성과 오퍼레이션들은 또 다시 속성 객체로 표현이 된다.

[그림2-2]의 개체 집합의 개념을 이용하여 각 클래스들의 속성을 확인해보자. 확인 절차는 객체 식별 절차와 같아서 문제 설명서로부터 명확히 식별하여 가려내게 된다.

4.2.3.1 설비클래스

“A”라는 설비명을 가진 설비 클래스는 설비코드로 식별되고, 총물량만큼의 자재를 총공수의 입력으로 주어진 제작기간 내에 만들어지게 된다. 이러한 문제 설명은 바로 [그림4-7]의 “A”에 나타난 개념적 모형을 기술한 것이다. [그림4-7] A는 객체의 속성중에서 오퍼레이션 부분에 대한 표기가 빠져 있다. 이러한 문제를 해결해 주는 표기법이 바로 OMT인 것이다. “B”에 OMT을 사용한 설비 클래스를 도시하였다. 계속되는 클래스 속성의 식별에는 A에는 제시하지 않고, OMT만을 제시하기로 한다. 그러나 OMT에 표현된 속성들은 속성 객체 혹은 관계 객체로서의 식별과 평가 과정을 거친 최종 산출물이다.



[그림4-7] 설비 클래스의 속성

4.2.3.2 표준장치 클래스

‘AA’라는 장치명을 가진 표준장치 클래스도 장치 코드로 식별된다. 해당 장치의 물량, 공수 및 제작기간은 바로 한 단계위에 있는 상위 계층의 단

위를 기준으로 하는 구성비로 표현되고, 전체 설비의 구축 혹은 조립 순서에 따라 장치간의 선후행이 나타나므로 이 장치 다음에 연결될 다음 장치와 다음장치에 연결될 위치를 연결될 위치의 작업 기간을 100으로 본 가중치로 갖는다. 또 각 장치마다 설계부서와 제작 부서를 속성으로 갖는다.

4.2.3.3 수주장치 클래스

표준장치라는 것은 특정 설비에 대한 표준적인 장치 모두를 일컫는다. 수주장치는 표준장치로부터 수주된 설비의 명세에 맞는 장치들을 추출해낸 상태를 말한다. 따라서 표준장치에는 포함되나 수주되지 않는 장치가 있을 수 있다는 것이다 이는 사례 기업의 관계형 데이터베이스 스키마 도출 과정에서 발견된 관계형 모델의 확장 요구의 증핵을 이루는 것으로, 객체 고유의 오퍼레이션을 객체의 데이터를 묶어 보려는 캡슐화의 시도로 나타나게 된다.

수주장치의 표현에는 공사 번호와 장치 코드가 함께 나타난다. 즉, 특정 공사에 대해 표준장치로부터 선정되었다는 의미를 내포하게 된다. 통합 관리 시스템에서 요구하는 것은 이러한 수주장치의 공수, 물량, 제작기간이 실제로는 얼마이며, 실제 후행장치와 연결위치가 어디냐는 것이다. 이러한 요구는 객체 ‘수주장치’가 공수, 물량, 제작기간, 그리고 연결위치를 산출하는 오퍼레이션과 실제 후행장치를 추적하는 오퍼레이션 속성을 가지는 것으로 객체 모델에 표현될 수 있다.

4.2.3.4 표준생산단위의 클래스

표준생산단위는 이미 설명한바 있는 표준장치와 같은 맥락에서 명세될 수 있다. 표준장치가 설비에 대해 종속되는 것처럼 표준생산단위는 한 장치에

대해 종속되어 그 장치를 구성하는 표준적인 하위시스템이 되는 것이다. 따라서 설비에 대해 표준장치가 그렇듯이 장치 대한 표준생산단위 역시도 집단화 관계로 나타난다. A라는 이름을 가진 표준생산단위는 생산단위코드에 의해 식별되고, 공수와 제작기간 및 연결위치를 가중치 값으로 가지며, 전체 장치의 구축이나 조립 순서에 따른 선후행을 표시하는 다음 단위를 표시하게 된다. 사례기관의 물량에 대한 관리는 장치 수준에서만 이루어지므로 생산단위부터는 물량에 관계되는 속성들은 불필요하다. 어느 부서에서 설계제작 하는 지에 대한 정보는 역시 포함하게 된다.

4.2.3.5 실제생산단위 클래스

실제생산단위는 표준생산단위들 중에서 수주의 결과로 수주장치에 포함된 생산단위만을 일컫는다. 수주장치와 표준장치간의 관계와 같이 실제생산단위와 표준생산단위도 공수와 기간 및 연결위치를 산출하고, 실제 후행단위를 추적하는 오퍼레이션을 속성으로 가지게 된다.

4.2.3.6 표준공정단위 클래스

‘AA’라는 공정명을 가진 표준공정단위는 역시 공정단위 코드에 의해 식별된다. 공수와 제작기간에 대한 가중치와 다음 공정에 대한 정보를 포함하는 것은 표준생산단위와 동일하나 설계는 생산단위를 기준으로 이루어지므로 실제 부서에 대한 정보는 불필요하며, 연결위치는 무조건 다음공정의 첫머리에 붙는 것으로 단순화시켰으므로 별도로 나타낼 필요가 없다.

4.2.3.7 실제공정단위 클래스

실제 생산단위의 속성들 중에서 연결위치에 관련된 속성만을 제외한 형태가 된다. 표준공정단위 클래스에서와 같이 생산단위 수준에서만 도면과 자

재에 대한 관리가 이루어지고, 공정단위의 후행연결은 직렬화 되는 것으로 가정한다.

4.2.3.8 도면 클래스

“D”라는 도면명을 가진 도면은 도면 코드에 의해 식별되고, 그 용도와 크기 그리고 설계 공수와 리드타임에 대한 정보를 포함한다.

4.2.3.9 도면계획 클래스

도면계획 클래스는 실제 생산단위와 도면 클래스의 관계 객체로서 두 클래스의 속성을 참조하여 출도일을 계산하는 오퍼레이션을 통해 출도일을 속성으로 가진다.

4.2.3.10 자재군 클래스

“M”이라는 자재군명을 가진 자재군은 자재군 코드로 식별되고, 계약기간, 제작기간, 그리고 운송기간을 정보로 가진다.

4.2.3.11 자재군계획 클래스

실제생산단위와 자재군 클래스간의 관계 객체로서 두 클래스의 속성을 참조하여 어떤 자재군의 입고일이 언제인가에 대한 정보를 포함한다.

4.1.3.12 일정계산클래스

일정계산 오퍼레이션을 통해서 입력된 정보를 구체적인 일정으로 출력하는 계산 모듈을 말한다.

각 클래스의 오퍼레이션들 중에서 주의해야 할 것은 바로 표준장치 혹은 표준생산(공정)단위가 수주의 결과에 따라 수주장치 혹은 실제생산(공정)

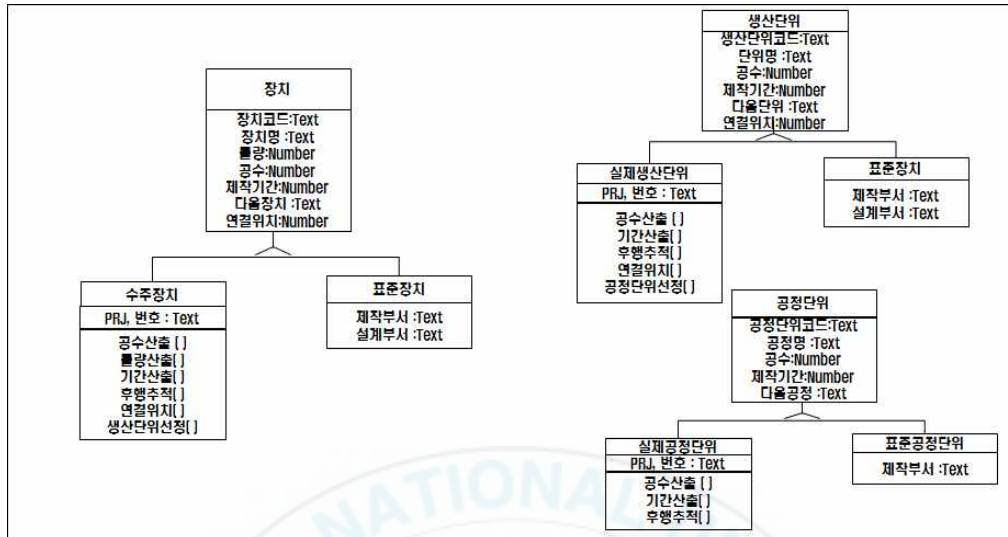
단위에 포함시키는 오퍼레이션이다. 이미 언급한 바와 같이 표준 데이터베이스는 한 설비에 관한 모든 관련 장치 및 단위에 대해 구성되며 표준을 위한 가중치의 부여는 바로 한 단계 위의 상위 수준을 기준으로 하는 구성비로 결정된다.

따라서 수주 시에 납기와 설비 규모에 대한 정보가 주어지면 표준적인 일정계획을 산정할 수가 있다. 그러나 주문 설비의 특성상 수주단위가 다양하여 특정 장치만을 수주했다거나, 한 설비의 표준장치 중에서 몇 개만을 수주했다고 할 경우 표준 데이터베이스의 가중치는 완전한 구성비일수가 없다. 이 문제를 해결하기 위한 오퍼레이션은 표준장치나 단위 중에서 수주된 것만을 포함하는 장치 구성비를 실행 시에 제시하게 된다.

이제까지의 기술을 바탕으로 속성들을 모두 포함한 클래스와 클래스간의 관계를 [그림4-8]에 도식한다. 클래스의 속성들 중에서 밑줄로 표시된 부분은 가중치를 속성 값으로 갖는 것들이다.

4.2.4 일반화를 통한 객체 클래스의 단순화

일반화는 객체들에 있어 공통적인 성질들을 상위 객체로 정의하고, 특수화된 객체들을 하위의 부분형 객체로 정의하는 추상화 방법으로 객체들간의 IS_A 관계를 나타나게 된다. 특수화란 일반화의 개념과 같으나, 클래스를 보는 시점에 있어 상위의 클래스에서 하위의 클래스를 보는 관점으로 하위 개념으로 내려갈수록 인스턴스가 특수화된다. [그림4-8]에 있어서는 일반화를 이용하여 추상화함으로써 관계도를 단순화 할 수 있는 여지를 발견할 수 있다. 그것은 표준장치와 수주장치, 표준생산단위와 실제생산단위, 그리고 표준공정단위와 실제공정단위 클래스를 각 장치, 생산단위, 그리고 공정단위라는 상위 클래스를 개입시켜 일반화함으로써 가능하다.



[그림4-9] 일반화 관계의 적용

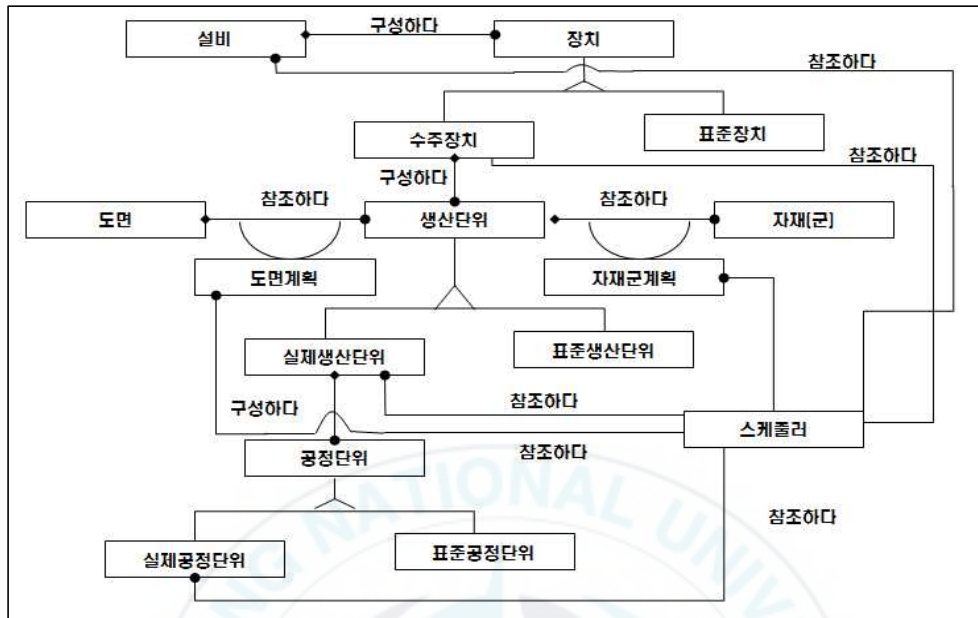
따라서 일반화 관계를 통해 단순화된 객체 클래스 관계도는 [그림4-10]과 같이 나타나게 된다.

4.2.5 데이터 모델의 도출

4.2.1에서 4.2.4의 과정을 통해 제시된 객체 모델은 데이터 모델 도출에 그대로 이용될 수 있다. 단, 데이터 모델의 설명력을 높이기 위해 기본 클래스가 아닌 복합 클래스를 자료형을 갖는 경우는 화살표를 이용하여 표기하는 것이 추천 될 만하다.

[그림4-11]은 도출된 데이터 모델을 도시한 것이다.

객체 모델에서 데이터 모델로 변환되는 과정을 살펴볼 필요가 있다. 물론 이 사례에서의 예가 일반성을 갖는 것으로 보기에 무리가 있겠으나, 하나의 준거는 될 수 있을 것이다. 첫째는 [그림4-9]의 일반화 관계를 기반으로 수주장치와 표준장치를 장치로, 실제생산단위와 표준생산단위를 생산단



[그림4-10] 일반화 관계를 포함한 객체 클래스 관계도

위로, 그리고 실제공정단위와 표준공정단위를 공정단위로 나타냈다는 것이다. 이는 일반화 관계란 분명히 별개의 객체가 집단을 이룬다는 특수한 연관관계에 의해 발생하는 집단화 관계와는 달리 사실상 동일한 한 객체에 대한 관계라는 표현이라는 측면에서 어쩌면 당연한 것일 수도 있다. 또한 하위 클래스 각 각을 모두 하나의 독립된 개체로 표현하는 것은 데이터 모델을 복잡하게 할 수가 있다.

둘째는 객체라는 것이 실체성을 가진 모든 개체를 그 대상으로 설계되므로 데이터 모델 내에 포함될 개체로는 부적합한 것들도 있을 수 있다는 것이다. 당 사례의 경우 “일정계산” 이라고 하는 객체가 거기에 해당된다 하겠다. 데이터 모델의 견지에서는 일정계산 객체에게 메시지를 전달하는 하나의 오퍼레이션만을 추가함으로써 그것을 일단 논외로 할 수 있을 것이다. 따라서 설비 클래스에 착수일 산출과 완료일 산출이라는 오퍼레이션을

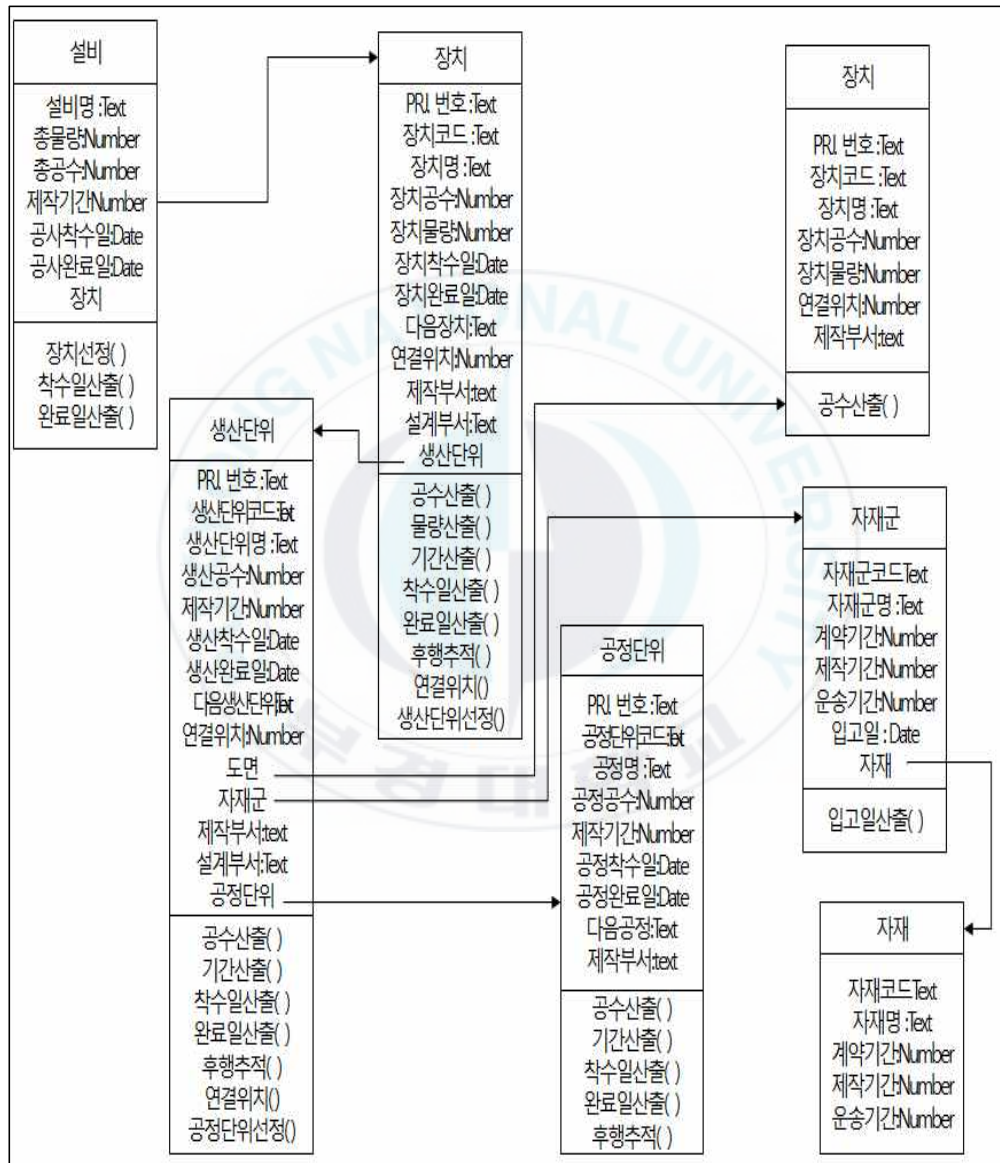
추가함으로써 일정계산 객체에게 주어진 자료에 의해 구체적인 착수일과 완료일을 산출해 달라는 메시지를 전달하게 할 수 있다. 나머지 장치, 생산단위, 공정단위, 도면, 자재군 등의 클래스에도 동일한 원리에 따라 구체적인 날짜의 산출은 일정계산 객체에 메시지를 보내는 오퍼레이션을 추가 하겠다. 이는 설계의 대상이 되는 문제영역에 존재하는 개체라면 그것이 데이터가 되었던 처리절차가 되었던 혹은 개념적인 개체이든 구체적인 개체이든 가뭇 그 모두를 객체로 파악하여 분석과 설계를 가능하게 하는 객체지향 방론의 또 다른 장점을 이용한 것이다.

4.3 적용기업 데이터 모델에 대한 논의

4.3.1 적용기업 데이터 모델의 특징

대형설비를 수주생산하는 적용기업의 생산 일정계획용 데이터 모델을 통해 단속적 생산시스템의 일정계획과 같은 맥락에서 대형설비 생산시스템의 일정계획이 관리될 수 있다는 것이 확인된다. 그러나 그 규모와 생산기간에 따르는 특성상 프로젝트 형태의 생산시스템과도 유사한 성격을 가지게 되는데, 대형설비 생산시스템의 일정계획용 데이터 모델의 특징들은 [그림 3-10]의 클래스 관계도와 [그림3-11]객체지향 데이터 모델을 [그림 4-10]의 클래스 관계도와 [그림4-11]의 데이터 모델과 비교함으로써 확인될 수 있다. 제3장에서 설계된 객체모델과 데이터 모델은 고객주문에 의해서 생산활동에 착수하게 되는 단속적 생산에 대한 일정계획 데이터 모델로 볼 수 있다. 대형설비의 생산 역시도 고객 주문을 수주함으로써 생산 활동이 시작되나, 이미 밝힌 바와 같이 하나의 프로젝트와 같은 성격을 갖게 된다.

따라서 [표3-1],[표3-2],[표3-3]에 제시된 프로젝트 생산시스템의 관리와 절차를 고려함으로써 대형설비 생산 일정계획에 대한 데이터 모델링이 가능하다.



[그림4-11] 도출된 데이터 모델

수주된 하나의 설비를 장치와 생산단위, 그리고 공정단위 등의 클래스로

나누어 각 단위의 공수나 착수 및 완료일 등을 그 클래스의 속성으로 취한 것은 프로젝트 일정계획에서 프로젝트를 여러 과업과 부분과업 등으로 나누어 각 활동 단위별로 일정을 수립하는 것과 유사하다. 그러나 전개된 각 하위단위에 대해서 참조할 만한 도면 정보나 자재 정보를 연계시켜 주기 위해 자재와 도면 정보를 참조할 데이터베이스는 존재하지 않는다. 단속적 생산시스템의 일정계획의 성격을 나타낸 것이다. [그림3-11]과 그림 [4-11]의 데이터 모델에서는 공통적으로 착수일과 완료일을 산출하는 오퍼레이션이 많이 나타나고 있는데 그것은 객체 모델에서 나타났던 일정계산(스케줄러) 클래스의 기능들을 객체지향 데이터 모델의 캡슐화 개념을 통해 해당 클래스의 오퍼레이션으로 표현한 것이다. 또한 각 클래스들의 연관 관계가 내포된 클래스의 형태로 데이터 모델 오퍼레이션상에 나타났다.

적용기업의 객체 모델과 데이터 모델에서 나타난 표준장치, 표준 생산단위, 그리고 표준 공정단위 등의 클래스는 표준 자료를 활용하여 실제 수주 설비의 BOM이나 소요 도면을 사전에 준비 내지는 참조하려고 시도한 것이다. 이는 마치 기성양복점에서 몇 몇 대표적인 치수의 양복들을 생산하여 소비자에게 조달하는 것과도 비슷한 양상을 취하게 된다.(물론 재고를 보유하고 있지는 않지만) 그에 따라 각 클래스에는 표준자료(가중치로 주어짐)로부터 수주 명세에 다른 실제 값으로의 변환을 위한 몇 몇 오퍼레이션들이 추가되고 있는데, 공수산출이나 물량 및 기간산출 등의 오퍼레이션들이 그것이다. 또한 하위 단위 생산 활동들의 독립성이 강하여 표준 사양에 속해 있는 많은 하위 활동들이 실제 수주 사양에서는 제외 도리 수도 있는 상황이 발생하게 된다. 이에 따라 어떤 하위 단위는 포함하고 어떤 하위 단위는 제외하는 오퍼레이션과 그에 따른 후행 활동을 추적하는 오퍼레이션이 각 각 추가되었다. 그리고 도면과 자재의 관리를 생산단위라는

활동에서만 수행하기로 함에 따라 도면과 자재 클래스에 대한 참조 관계가 많이 줄어들었다.

또한 표준 장치와 수주 장치, 표준 생산단위와 실제 생산단위, 그리고 표준 공정단위와 실제 공정단위에 있어서는 일반화 관계를 통하여 객체 모델을 단순화 할수 있었다. 이를 통하여 객체지향 시스템에서는 클래스의 속성과 오퍼레이션의 계승되므로 시스템이 확장이 용이해진다. 그러나 일반화 개념은 집단화와는 달리 동일한 클래스에 대한 개념적 구분이므로, 상위 클래스와 하위 클래스를 하나의 클래스로 표현할 수도 있다. 같은 맥락에서 [그림4-11]의 데이터 모델에서는 [그림4-10]의 객체 모델과는 달리 표준 활동과 실제활동간의 구분이 없이 단일 활동으로 표현되었다. 이는 객체지향 개념에서 볼 때, 가중치 값을 가지는 표준 활동과 실제값을 갖는 실제활동은 동일한 클래스에서 인스턴스화된 하나의 객체 인스턴스일 뿐이기 때문이다. 객체지향 데이터베이스 관리시스템을 사용하게 될 경우 (OID)를 가진 한 클래스에서 인스턴스화된 객체로서 존재하게 될 것이다. 이렇게 도출된 대형설비 생산 일정계획을 위한 객체지향 데이터 모델은 제3장의 일반적인 생산 일정계획의 데이터 모델과 유사한 클래스들과 그들간의 관계에 의해서 구성된다. 물론 각 클래스의 구체적인 속성과 오퍼레이션들에는 차이가 있을 수 있다.

4.3.2 적용기업 데이터 모델의 시사점

적용기업이 객체지향 데이터베이스 시스템을 구축하는 것이 아님에도 불구하고 객체지향 데이터 모델의 설계를 시도한 것은 객체지향 데이터 모델은 관계형 데이터 모델도 포함하는 보다 폭 넓은 표현력을 가진 데이터 모델이라는 점에서 설명될 수 있다. 데이터베이스 관리시스템이라는 것은 다

를 아닌 데이터 모델에 표현된 개념적 측면을 물리적으로 뒷받침하기 위한 소프트웨어이기 때문이다.

시스템을 개발하거나 개선함에 있어 문제가 되는 것 중 하나가 기존 시스템의 재사용과 여러 연구개발팀의 최종산출물을 통합하는 문제일 것이다. 객체지향 개념은 그 내재적 특성에 비추어 매우 바람직한 방향으로 시스템 개발의 과정을 이끌어 가게 된다. 그러한 내재적 특성은 데이터 추상화, 객체캡슐화, 계층 그리고 다형성 등을 말한다. 이를 통해서 소프트웨어 엔지니어링에서 요구되는 추상화, 모듈성, 모듈내 응집력, 정보은닉과 같은 원칙들이 자연스럽게 지원된다[18].

적용 기업의 예에서 객체지향 데이터 모델을 설계함으로써 얻어지는 효익을 몇 가지고 정리할 수 있다.

첫째 객체 모델링의 강력한 표현력을 기반으로 데이터 모델의 의미적 측면이 논리적으로 명확하게 표현되었다는 것이다. 데이터베이스 개발 과정에서 데이터 모델이란 문제 영역에 대한 개념적 설계에 해당된다. 관계형 데이터 모델에 비해 표현력이 뛰어난 객체지향 접근법의 객체 모델은 일반 사용자가 문제 영역에 대해 느끼고 있는 관념을 그대로 객체를 통해서 모형화하게 되므로 의미적 측면에 대한 지원이 확실하며, 데이터 모델로의 전환도 손쉽게 이루어질 수 있다. 사용자가 관심을 가지는 것은 설비나 장치에 대한 정보가 저장된 릴레이션이 아니라 자신이 일정을 수립해야하는 설비나 장치 그 자체인 것이다. 객체지향 시스템은 그것을 가능케 해주며, 객체모델이나 객체지향 데이터 모델이 출발점이 되는 것이다.

둘째, 개발팀이 서로 다른 도면과 자체관련 자료들도 설비나 장치, 혹은 생산단위 클래스 내에서 접근할 수 있게 됨으로써 기 개발된 산출물에 대한 통합의 효과를 얻을 수 있다. 하나의 설비에 대한 모든 정보가 복합객체와 내포된 객체에 의해 깔끔하게 되므로 우수한 사용자 지원이 가능하게

되고, 시스템 개발자와 개발팀간의 의사소통에도 도움이 된다. 객체로서의 설비에만 주문할 수 있게 해주는 것은 객체를 중심한 철저한 모듈성과 응집력에 기초하여 얻어진 효익이다.

셋째는 생산 일정계획의 수립에 관한 것이다. 생산일정 중에서 대일정계획은 장치 클래스에 대해서, 중일정계획은 생산단위에 대해서, 소일정계획은 공정단위에 대해서 이루어지게 된다. 이 때 클래스들은 일정계획에 사용될 자료형과 자료값 뿐만이 아니라 그것들을 조작하여 구체적인 일정을 수립하는 오퍼레이션들도 같이 정의되고 캡슐화되어 있으므로, 대중소일정계획이 해당클래스 수준에서 수립된다.

넷째는 시스템 개발의 확장성과 유연성에 대한 효과이다. 객체라는 것은 그 대상이 데이터베이스일 수도, 내부 모듈일수도 사람일수도 있다. 따라서 제4장을 통해 도출된 객체 모델은 전체 시스템을 객체지향 시스템으로 개발할 수 있는 기반이 될 수 있다. 새로운 객체가 소개되었을 경우 그 속성과 오퍼레이션을 정의한 후 객체 클래스 계층도에 삽입하여 다른 객체와의 연관관계를 표현해 줌으로써 시스템 설계와 개발이 진행될 수 있다는 것이다.

관계형 데이터베이스를 기반으로 하는 객체 지향 시스템의 개발과 관련된 사례들에 대한 연구도 많이 발견되고 있다. 객체지향 방법론이 프로그래밍 분야에서만 사용되는 개념이 아니라 시스템 분석·설계·구현에 걸쳐 폭넓게 사용되고 있다는 것을 생각한다면 당연한 것이다.

끝으로 적용기업의 경우 표준 데이터베이스 외에도 실적공사 데이터베이스가 유지되고 해당 데이터베이스에 대한 새로운 버전들이 계속 발행한다면, 장기적으로는 데이터베이스의 버전 기능을 지원하도록 개발되고 있는 객체지향 데이터베이스 시스템으로의 전환이 예상될 수 있다. 따라서 객체지향 데이터 모델의 설계는 상당히 미래지향적인 의미를 가지고 있다.

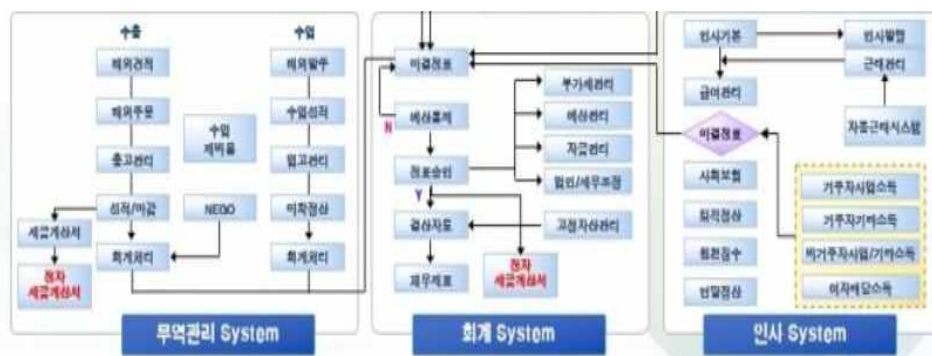
V. 활용시나리오

5.1 시나리오 설정

본 장에서는 본 논문의 대상인 (재)부산테크노파크는 고객사로부터 발주 정보를 받아 생산가공을 하고 가공 되어진 시제품을 실험 테스트 후 인증 절차를 거쳐서 고객사에 납품하는 공공기관 연구소이다. 고객사의 주문에 의한 수주생산방식과 자체 기획모델을 대상으로 생산계획을 수립 생산하는 재고생산방식의 제조 환경을 가지고 있다.

따라서 거래처별 예상판매계획, 및 제품사양정보, 납기일 관리뿐만 아니라 완제품 재고 및 수주대비 출고현황 등의 전체적인 영업관리 정보가 실시간으로 처리되며 고객의 요구사항 변동에도 신속 정확한 의사 결정을 할 수 있는 정확한 정보가 필요하다.

현행 부산테크노파크사의 ERP 시스템 구성도는 [그림 5-1]과 같이 ERP 서버를 운영하며 ERP서버에 접속하여 크게 3가지 카테고리로 업무를 수행하고 있다.



[그림5-1 현재 ERP구성도]

따라서, 전표입력만으로 자금, 예산, 부가세, 법인조정 업무와 실시간 연동 처리되고 과거 전표를 월별/일별/결의부서별/작성자별/내역별로 검색하여 복사 가능한 중복방지업무를 수행할 수 있으며, 기간별 주요 관심분야 증감 잔액보고서를 사용자정의에 의해 작성 가능하고 인사조직의 전 과정도 통합 관리되고 있다. 그러나 더욱더 체계적인 정보시스템을 구축하기 위해서는 장소별 입고처리 및 실시간 재고 확인 가능하고 입고부터 출고까지 개별법 프로세스를 통한 재고추적관리기능 지원 및 다양한 재고 평가 방법 및 수불통제 기능 제공할 수 있는 일정계획시스템도 추가 구축될 필요가 있다. 추가 통합 구축시 공정부터 다음 공정까지 생산관리 환경에 맞는 프로세스 적용을 가능토록하고 자재의 정확한 실시간 제공관리도 지원 가능하고 각 고객사의 기업환경에 적합한 공정관리부터 다양한 BOM관리 지원으로 복잡한 생산일정계획 관리도 가능할 수 있을 것으로 기대된다.

5.2 시나리오 설정

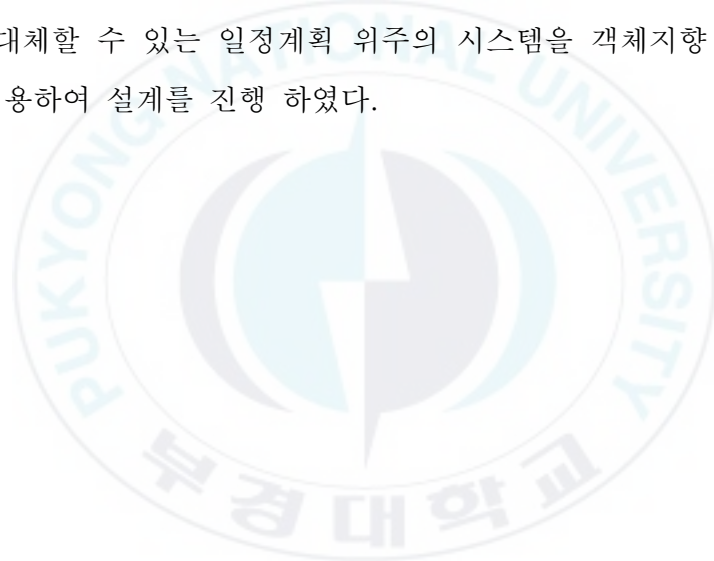
본 연구에서는 영업관리 AS-IS 프로세스 분석하고 완제품 재고 및 수주 대비 출고현황 등의 영업관리 정보가 실시간으로 처리되며 고객의 요구사항 변동에도 신속 정확한 의사 결정을 할 수 있도록 TO-BE 프로세스를 설계하였다.

최종제품이 납기에 생산될 수 있도록 여러 구성품과 부품을 생산하고, 그때 소요되는 자재와 도면에 대한 일정계획들을 구매시스템, 생산시스템, 영업시스템 모듈을 추가 처리함으로써 수작업 처리함으로써 겪는 업무의 비효율성을 극복 하기위한 시스템개발 설계를 하였다.

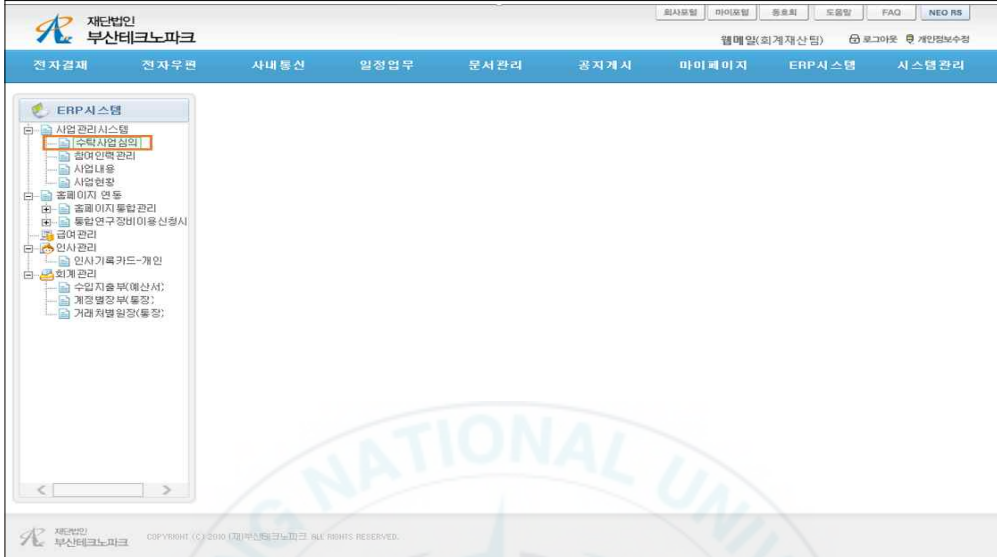
[그림 5-2]는 본 논문의 연구 대상인 부산테크노파크의 ERP 시스템의

전체적인 향후 흐름을 도식화 한 것으로서 현행 생산일정계획의 주요 제점으로는 예상판매 계획이나 수주에 따른 납기 일정관리 및 수주대비 납품실적에 대한 관리가 미흡하였다. 또한 담당자는 대부분의 정보를 수작업으로 관리하거나 비효율적인 시스템 기능을 사용함으로써 고객사 및 자사의 완제품 재고 현황에 대한 파악이 어려워 수주에 따른 미납현황과 생산일정에 따른 입고 예정정보 및 반품 정보에 대한 관리가 제대로 이루어지지 못하였다.

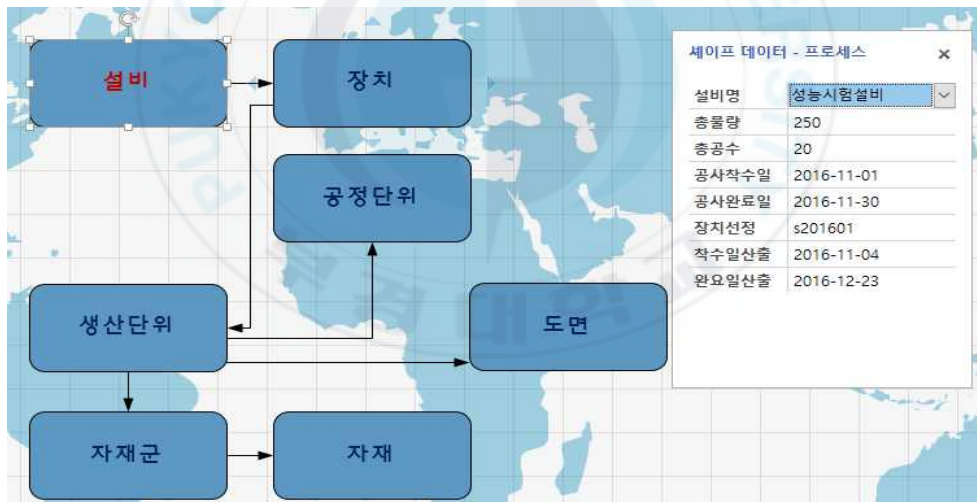
본 논문에서는 일정관리 현행업무를 상세 분석하고 고객의 요구사항에 능동적으로 대체할 수 있는 일정계획 위주의 시스템을 객체지향 방법론으로 OMT를 이용하여 설계를 진행 하였다.



[표 5-1] ERP 설비정보화면



1. 화면설명



ERP시스템의 메뉴 구성도이며, 사업관리시스템은 좌측메뉴의 "사업관리시스템" 부분임이다.

좌측메뉴에서 "설비", "장치", "생산단위", "공정단위", "생산단위", "자재군", "자재", "도면"의 7개의 서브메뉴로 구성된다.

2. 주요기능 키

- 담당직원의 로그인 가능한 ERP시스템에서 담당팀별로 입력한다.

[표 5-2] ERP 장치 및 생산단위, 자재군 등록화면

세이프 데이터 - 프로세스.85

장치	성능시험설비
총물량	250
총공수	20
공사작수일	2016-11-01
공사완료일	2016-11-30
장치선정	s201601
착수일산출	2016-11-04
완료일산출	2016-12-23

세이프 데이터 - 프로세스....

자재군코드	MA_CNGHX01
자재군명	DUCT_210
계약기간	2016-11-04
제작기간	2016-11-10
운송기간	2016-11-05
입고일	2016-12-06
입고산출일	2016-11-09

세이프 데이터 - 프로세스.129

PRT번호	S201601
생산단위코드	CNGHX01
생산단위명	CNGHX02
생산공수	20
제작기간	14
생산착수일	2016-11-02
생산완료일	2016-11-23
다음생산단위	CNGHX03
연결위치	201
도면	DO_CNGHX01
자재군	MA_CNGHX01
제작부서	생산팀_이재운
설계부서	도면팀_신성기
공정단위	2016-11-08
공수산출	2016-11-11
기간산출	2016-11-18
후행수직	2016-11-21
연결위치	VALVE03
공정단위선정	GOAL_S201601

1. 화면 설명 - 생산단위에 따른 자재 정보를 담당자가 입력 조회 가능하다.

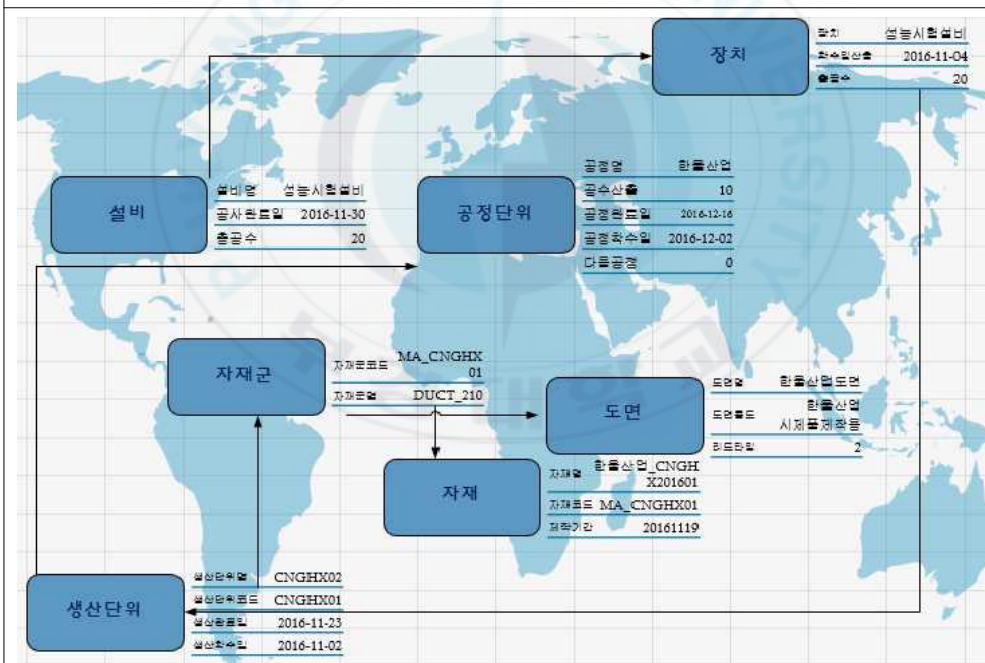
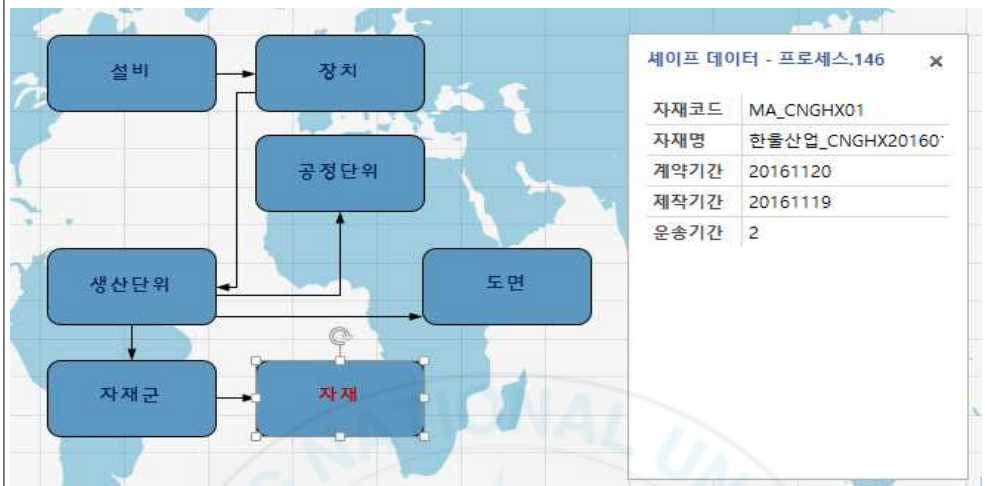
2. 주요기능 키

- 생산단위에 따른 자재의 진행상태를 파악할 수 있다.

[표 5-3] ERP 공정단위 및 도면(군)



[표 5-4] ERP 자재 및 대시보드



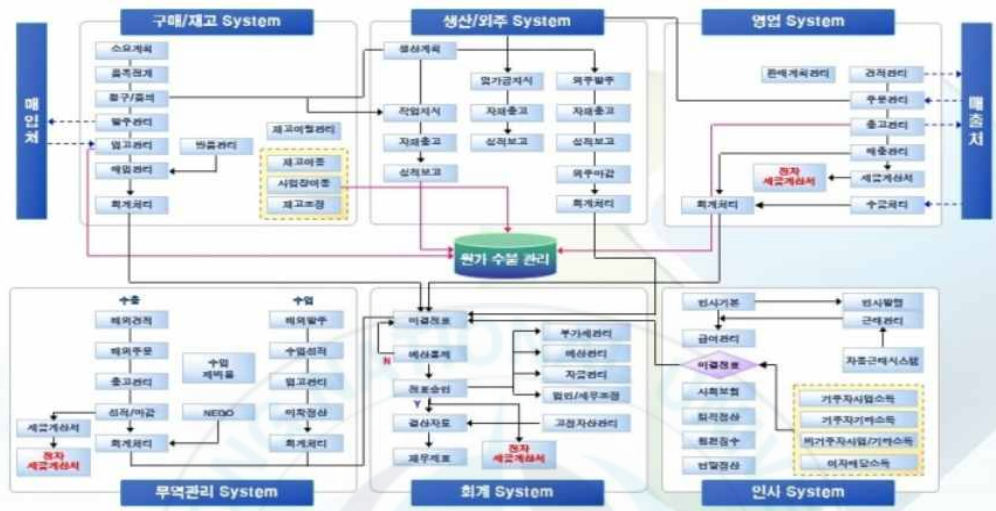
1. 화면 설명

- 필요한 자재의 재고상태를 파악할 수 있다.
- 년도/월별,센터별,부서별,팀별 그래프 및 그리드로 출력되는 화면이다.

2. 주요기능 키

발주서에 대한 입출고 및 재고 이력관리, 입출고 현황을 바탕으로한 외주 거래처 재고 관리 등 한눈에 파악가능한 대시보드 지원

5.3 기대효과



[그림5-2] 향후 ERP 설계도

[그림 5-2]은 주요 고객사로부터 수주 정보를 이용하여 고객사의 송신 정보를 수신 받아서 내부 ERP 시스템으로 처리하는 프로세스 흐름도 이다. 본 논문에서 재설계한 TO-BE 프로세스는 AS-IS 프로세스를 기반으로 생산일정관리 업무의 목표를 달성하기 위하여 고객사로부터 수신한 판매계획 정보, 생산계획정보, 발주정보, 입고정보 등의 내역관리를 체계적으로 하고 정보를 수신 후에 내부 프로세스에 영향을 주는 부분에 대한 처리를 명확하게 함으로서 고객사에게 판매계획에 대한 납기응답, 발주정보 (Purchase Order)에 대한 납기응답 등의 정보를 정확하게 수행할 수 있는 시스템으로 설계하였다. 또한 수주대비 출고, 출고대비 고객사의 입고 실적관리, 납품실적 대비 수금현황 프로세스를 영업관리 시스템에 반영함으로써 기존 업무를 개선하고자 하였다. 또한 기존의 프로세스에서는 각 부서 내에서만 활용되던 정보를 타부서와 공유할 수 있도록 개선함으로써

전체 생산일정관리 업무와 연계된 생산 및 자재부서에서는 정보를 보다 신속하고 정확하게 처리할수 있고 제조, OPTION BOM 등 다양한 형태의 기준정보관리 가능작업지시 품목에 대하여 공정/작업장 별로 자재청구 기능 지원 및 제품검사에 대한 다양한 분석현황 제공까지도 기대해본다.

따라서 부산테크노파크는 기업이 상대적으로 장비의 구축이 어려운 기업에 시험/검사 기회가 확대되어 새로운 제품 개발 및 품질 경영으로 인한 대고객 서비스 기회를 확충할 수 있어, 국내 기업진흥에 직간접적으로 영향을 발휘하여 국내 기업에 다양한 품질 검사 기반을 제공하여 국내 생산 제품의 품질수준 향상이 가능토록 지원 할 것이다.

기술 인프라의 구축과 관리는 기술이 무기가 되는 21세기 글로벌 환경에서 매우 중요한 국가적 과제이며, 기술 인프라의 중요한 요소 중 하나인 시험·분석·연구 장비의 전략적 운영은 이러한 맥락에서 중요한 의미를 갖는다. 시험·분석·연구 장비 전략적 운영 방법의 하나인 장비의 활용은 가동률 증가로 인한 경제성의 제고로 시험·분석 당사자에게 이익을 제공할 수 있으며, 국가기술 발전의 계기를 제공하고, 기업을 진흥시킬 수 있는 방안이 되고 기술 인프라의 균형과 확대를 도모할 수 있어 지역균형 발전 정책을 위한 부산테크노파크 정책에도 부합되리라 예상된다.

VI. 결론

6.1 논문의 요약

비즈니스 프로세서를 설계하는 것은 궁극적으로 이를 소프트웨어로 구현하기 위한 사전단계로 인식된다. 따라서 본 연구에서 수행된 프로세서에 따라서 객체 지향적 시스템을 구현하는 것이 궁극적인 목표 달성이 될 것이다. 지금까지 대형설비 생산 일정계획용 데이터베이스의 개념적 설계를 위해 객체지향 방법론의 객체 모델링 과정을 통해 객체지향 데이터 모델을 도출하는 것에 대해 살펴보았다. 최근까지 상용 데이터베이스 시장을 관계형 데이터베이스 시스템의 석권하다시피 한 상황에서 관계형 데이터 모델의 사용은 불가피한 면도 있었다. 그러나 관계형 데이터 모델은 분명한 한계점을 가지고 있다. 관계형 데이터 모델은 데이터베이스 스키마로의 전환을 용이하게 하겠다는 목표 하에 이른바 데이터 모델의 해석 원칙만을 강조한 방법론이라 할 수 있다. 따라서 관계형 데이터 모델로 표현되기만 하면 상용 관계형 데이터베이스 시스템으로 구현하는 것은 문제가 되지 않는다. 그러나 실세계의 엔터티들이 행/열의 간단한 릴레이션 구조 속에 완벽하게 표현되어 진다는 것은 불가능하다. 따라서 관계형 데이터베이스 관리 시스템의 효율을 높이기 위해 여러 의미적 측면에 대한 배려는 등한시되었다.

이러한 상황은 정보시스템 분야의 혁혁한 발전과 함께 도래한 최종사용자 컴퓨팅 환경의 추구라는 정보화 사회의 요구에 의해 더 이상 덮어 둘 수 없게 되었다. 모든 시스템이 일반사용자의 관념과 맥을 같이 함으로써 사용자 친화적인 정보시스템 환경이 조성될 때 비로소 가능하다. 따라서 일

반 사용자의 관념 내에 존재하는 많은 의미적 측면들을 등한시하는 관계형 데이터 모델에 대한 개선의 움직임이 일어날 수밖에 없었다. 의미적 측면을 강조한 의미 데이터 모델에 대한 연구가 없었던 것은 아니다. 그러나 의미 데이터 모델은 데이터베이스 시스템으로 상용화되지 못하였고, 관계형 데이터 모델 도출하기 위한 사전 작업 정도로 사용되어왔다.

그러나 최근 프로그래밍 분야에서 출발하여 시스템 분석·설계·방법론으로까지 그 영역을 확대하고 있는 객체지향 방법론과 객체지향 시스템은 데이터베이스 시스템 분야에서도 그 활용에 대한 기대가 높게 나타나고 있다. 특히 객체지향 데이터 모델은 표현력이 뛰어나 의미 데이터 모델이 추구한 의미적 측면에 대한 배려와 자료가 자료 조작용 오퍼레이션을 함께 정의할 뿐 아니라 객체지향 데이터베이스 시스템을 통해서 상용화가 가능하다는 점에서 상당한 기대를 일으키고 있다.

생산 일정계획을 수립하는 것은 최종 제품이 납기 내에 생산될 수 있도록 여러 구성품과 부품을 생산하고, 그 때 소요되는 자재와 도면에 대한 조달 일정을 수립하는 것이다. 이를 위해 최종제품, 구성품, 부품, 원자재, 도면, 고객주문, 생산능력, 스케줄러 등의 객체 클래스가 문헌 연구를 통해 식별되었다. 최종 제품, 구성품, 부품 클래스의 경우 각 각의 이름 공수, 제작 기간, 생산착수 및 완료일, 그리고 자재와 도면에 대한 정보를 속성으로 가지며, 연관된 다른 클래스의 속성값을 참조하여 공수나 일정을 계산하는 오퍼레이션을 가진다. 각 클래스들은 속성에 대한 자료형으로 문자열이나 정수형이나 날짜형 같은 기본 클래스들 외에도 원자재나 도면 혹은 구성품이나 부품과 같은 클래스 자체도 자료형으로 가짐으로써 복합 객체로 구성되고, 또 그들간에 내포관계가 발생한다[19].

대형설비 생산의 일정계획용 데이터 모델도 일반적인 경우와 비슷한 양상을 띤다. 대형설비의 생산은 일종의 프로젝트처럼 관리될 수 있으나, 한

사업부가 특정설비만을 지속적으로 생산하게 될 경우 1회성의 프로젝트와는 사뭇 다른 일정계획 수립과정이 나타난다. 대형설비를 수주하여 생산하는 중소기업을 대상으로 대상설비 생산을 위한 일정계획용 데이터베이스의 객체지향 데이터 모델을 설계하였다. 설비, 장치, 생산단위, 공정단위, 자재, 도면 등과 같은 클래스들이 식별되었고, 그들간의 관계가 클래스 관계도로 표현되었다. 이 클래스들은 일반적인 경우의 클래스들과 거의 유사하다. 다만 구체적인 속성과 오퍼레이션들에는 약간의 차이가 발생하였다. 속성의 경우는 하위 단위별로 일정계획에 소용될 자료 외에도 관리 목적상 요구되는 속성도(총물량)추가 되었다. 오퍼레이션의 경우는 표준화가 힘든 대형설비의 소요 자재나 설계도면 생산 일정계획의 범주에 포함시키고, 수주된 설비 사양에 따라 설비의 하위 장치나 생산단위가 선택적일 수 있음을 반영하기 위해 공수나 일정을 계산하는 오퍼레이션 외에도 생산하게 될 하위 단위를 선정하고, 선정된 하위 단위의 실제 후행 단위를 추적하는 등의 오퍼레이션이 추가 되었다. 그러나 기본적으로는 일반적인 경우의 데이터 모델과 동일한 클래스와 클래스 관계에 의해서 대형설비 일정계획 데이터 모델이 설계될 수 있었다.

또한 소요자재의 조달에 관련된 일정과 소요도면에 대한 일정만을 포함하는 데이터 모델을 필요에 따라 확장하여 그 외의 개념들이 객체 클래스로 정의되는 과정을 거쳐서 데이터 모델 내에 추가될 수도 있다. 예를 들면, 한 활동을 수행하는 작업팀이나 작업자에 대한 인사 관련 자료들도 작업자 혹은 사원이라는 객체가 도입됨으로써 데이터 모델에 추가될 수 있다는 것이다. 객체지향 시스템은 자기 충족성과 식별성을 가진 어떤 실체에 대해서도 객체 식별을 통해 시스템의 관리대상으로 삼게 되는데, 객체지향 데이터베이스는 다른 아닌 객체를 관리하는 데이터베이스 시스템인 것이다.

본 연구는 데이터 모델의 도출을 위해 객체지향 방법론의 객체 모델링을

활용하였다. 그 결과는 다름 아닌 객체지향 데이터 모델이나, 데이터 모델이라는 것은 그 정의 자체가 어떤 데이터 어떤 데이터베이스 시스템에 대해서도 독립적이므로, 도출된 데이터 모델을 기반으로 관계형 데이터베이스 시스템을 위한 스키마가 개발될 수도 있고, 객체지향 데이터베이스 시스템을 위한 스키마가 개발될 수도 있다. 물론 설계된 객체지향 데이터 모델은 객체지향 데이터베이스에 의해 구축되는 것이 가장 바람직하겠으나, 객체지향 데이터베이스에 대한 안정성이나 신뢰도에 대한 의구심이 아직 잔존하고, 특히 적용 기업이 관계형 데이터베이스 시스템을 기반으로 하고 있음을 감안할 때, 객체기반의 관계형 데이터베이스 관리시스템으로의 구현이 추천될 만하다. 관계형 데이터베이스와 C++같은 객체지향 프로그래밍 언어를 결합한 형태의 객체지향 데이터베이스 시스템이나 관계형 데이터베이스와 비주얼베이직 같은 객체기반 프로그래밍 언어를 결합한 객체기반 데이터베이스시스템의 개발에 대한 연구와 상용화에 대한 노력들이 계속되고 있다.

시스템을 분석·설계한다는 입장에서 객체지향 방법론을 통한 데이터 모델의 도출 과정은 데이터 모델링 이상의 부수적 효과들도 제공하게 된다. 객체지향 방법론을 통해 시스템을 개발한다는 것은 객체지향 시스템을 개발한다는 것을 의미한다. 객체지향 시스템의 당위성은 일반사용자가 가지고 있는 실세계 혹은 문제 영역에 대한 관념을 그대로 컴퓨터 시스템화한다는 것에서 찾을 수 있다. 객체라는 것이 바로 인간이 독립성을 가진 것으로 인식할 수 있는 구체적이거나 추상적인 그 무엇이므로, 그러한 객체를 중심으로 시스템을 개발한다는 것은 인간이 생각하는 대로 움직이는 시스템이 된다는 것이다. 따라서 가장 이상적일 경우 누구나 전산 전문가 없이도 컴퓨터 기반 시스템을 자유롭게 사용할 수 있게 될 것이다.

이미 도출된 객체 모델은 OMT에서 일반적으로 준비하게 되는 동적 모델

및 기능 모델과 합쳐짐으로써 이른바 객체지향 시스템의 개발이라는 또 하나의 성과를 얻을 수도 있다. 이는 데이터 모델의 개발이라는 당초의 목적 이외에 얻어지는 부수적인 효과 중에서 가장 큰 것임에 틀림없다.

이상의 연구에서 지적될 수 있는 한계점은 다음과 같다. 우선 데이터 모델의 도출 과정에서 지적될 수 있는 것으로서 객체 모델로부터 데이터 모델로의 전환 과정에서 적용될 수 있는 일반적인 규칙이 제시되지 못했다는 것이다. 둘째는 적용기업의 대형설비 생산 일정계획 데이터 모델 설계 과정에서는 데이터 모델이 완전히 시스템 독립적인 견지에서 도출되지 못했다는 것이다. 예를 들면 객체 모델링에서 자주 등장하는 표현인 한 클래스가 자기 자신에 대해 연관을 가지는 경우의 표현은 지양하였다. 이는 데이터 모델이 사용자에게 대한 설명력을 가져야 한다는 것을 생각할 때, 자기 연관 관계의 지나친 사용은 오히려 사용자를 혼란스럽게 할 수도 있기 때문이다. 셋째는 일반과 관계에 의한 속성과 오퍼레이션의 계승이나 다형성에 대한 명확한 입장이 제시되지 못했다는 것이다. 이 부분에 대한 처리 문제는 사실상 현재 개발중 이거나 개발되어 보급 중인 여러 객체지향 데이터베이스 관리시스템들에서도 공통되는 난점 중 하나이다.

대부분의 업무가 글로벌한 환경에서 수행되고 있으므로 본 시스템의 개발 플랫폼은 비용 및 효과면이나 정보기술의 개발 추세를 고려할 때 인터넷 기반이 가장 기대된다. 따라서 추후 웹을 기반으로 하고 분산된 환경에서 이용 가능하며, 소프트웨어의 재사용과 모듈화 및 조직과 환경변화에 신속히 적응 가능한 정보시스템 구축을 목표를 한다.

6.2 논문의 공헌

기술 인프라의 구축과 관리는 기술이 무기가 되는 21세기 글로벌 환경에서 매우 중요한 국가적 과제이며, 기술 인프라의 중요한 요소 중 하나인 시험·분석·연구 장비의 전략적 시스템 운영은 이러한 맥락에서 중요한 의미를 갖는다. 시험·분석·연구 장비 전략적 운영 방법의 하나인 장비의 공동 활용은 활용을 증가로 인한 경제성의 제고로 시험·분석 당사자에게 이익을 제공할 수 있으며, 국가기술 발전의 계기를 제공하고, 중소기업에 진흥시킬수 있는 방안이 되는 등 기술적, 사회적으로도 다양한 파급효과를 가져올 수 있다.

장비의 활용은 현재보다 장비의 가동률을 증가시켜 간접적으로 장비의 생산성을 향상시킬 수 있는 효과를 가져 올 수 있는데 현재 대상 장비의 외부 공동 활용은 전무한 상황으로 설문조사 결과 외부 공동 활용을 현재 활용의 45%까지 증가시킬 수 있는 것으로 나타나고 있다. 이는 현재 가동률 14%를 공동 활용을 통하여 장비의 가동률이 20.2%까지 증가할 수 있음을 의미하며 증가된 가동률에 비례하여 시험·검사 수수료로 인한 경제적 추가 이익이 발생할 수 있음을 의미한다.

산업계 기술기획 분야의 관리자로 구성된 전문가 집단의 장비 공동 활용에 대한 행정적 장점과, 경영상의 효과를 인터뷰 한 결과, 국내기관 및 기업의 기술력 향상으로 인한 국가 기술신인도 증대, 국가 기술인프라의 균형 발전, 정부 재정 운영에 대한 국민 신뢰도 향상 및 중소기업 진흥 등과 같은 다양한 사회적 효과를 유발할 것으로 분석 되었다.

6.3 한계점 및 극복방안

본 연구는 기존의 설계 및 개발 방법론과 다르게 업무분석, 객체지향시스템 설계, 향후 ERP의 다른 모듈에 대해 확장가능하고 재사용 가능한 객체지향 개발을 할 수 있는 기반 구축측면에서 논문의 기여도를 제시할 수 있다. 먼저 업무분석 측면에서 수요예측, 판매계획, 수불관리, 영업분석 등의 업무를 분석함에 있어 기존의 부정확한 업무 및 정보관리 환경을 개선하기 위하여 AS-IS 프로세스 분석과 TO-BE 프로세스 설계를 통해 기존 업무수행상의 문제점을 개선하였다.

둘째, 시스템 설계에 있어서 객체지향 방법론인 OMT에 의한 분석 및 설계를 수행함으로써 업무현상을 보다 정확히 모델링 할 수 있었으며, 단위 모듈 업무를 포함한 타 업무와의 모듈별 인터페이스를 정확히 도출 할 수 있었다. 특히 시스템 개발 측면에서는 객체지향기반의 시스템을 구현함으로써 개발기간을 줄이고, 개발자의 소프트웨어 개발 생산성을 향상하는 결과를 낳았다.

셋째, 구축 효과에 있어서 판매계획 및 판매 계획 대비 실적현황, 수주대비 출고현황, 고객사와의 정보연계 등 다양한 영업 분석정보를 경영자나 영업담당자, 그리고 타부서의 담당자에게 신속 정확하게 실시간으로 공유함으로써 비효율적인 수동 반복 작업을 줄이고 업무 처리 정확도를 향상하며 사용자에게 신뢰성 있는 다양한 형태의 정보를 제공해 줄 수 있게 되어 기업의 경쟁력 향상에 많은 도움이 되었다.

비즈니스 프로세서를 설계하는 것은 궁극적으로 이를 소프트웨어로 구현하기 위한 사전단계로 인식된다. 따라서 본 연구에서 수행된 프로세서에 따라서 객체 지향적 시스템을 구현하는 것이 궁극적인 목표 달성이 될 것이다. 대부분의 업무가 글로벌한 환경에서 수행되고 있으므로 본 시스템의

개발 플랫폼은 비용 및 효과면이나 정보기술의 개발추세를 고려할 때 인터넷 기반이 가장 기대된다. 따라서 추후 웹을 기반으로 분산된 환경에서 이용 가능하며, 소프트웨어의 재사용과 모듈화 및 조직과 환경변화에 신속히 적응 가능한 정보시스템 구축이 가능하다.

본 연구를 수행함에 있어서 한계점으로는 TO-BE 설계 시 고객사의 인터페이스(Interface) 수행 방법이 변경됨에 따른 기술적인 어려움이 있었고 개발적인 측면에서는 실제 구축되어진 자료에 대한 관련 문서가 부족하여 참조하기 어려운 점 등이 발생하였다. 또한 일반 기업체의 설비 생산일정 시스템에 대한 선행연구가 많지 않아 AS-IS 분석에 따른 TO-BE 모델의 설계 시 우수사례(Best Practice)에 대한 고려가 부족하였다. 이와 같은 한계점을 극복하고 향후 영업관리 외 타 업무에 대한 해당 중소기업의 표준 프로세스를 수립하고 OMT를 이용한 객체지향 방법론적 설계 및 구현이 필요하다.

또한 좀 더 나아가 인터넷 웹기반이 아닌 모바일을 이용한 수주정보 입력 및 배송정보 관리 등에 대한 연구가 필요하다

참 고 문 헌

- [1] J. A Campbell et al. "The Object-Oriented Design and Implementation of a Relational Database Management System", Journal of Object Oriented Programing. Jul-Aug. pp44
- [2] E. Nahouraii & F. Petry, Object-Oriented Databases, IEEE Computer Society Press, Chelsea(Michigan), pp.3
- [3] B. Henderson-Sellers&J.M. Edwards, "The Object-Oriented System Life Cycle" Communication of the ACM, vol33, Np.9 1990 pp. 147
- [4] 김원, “객체지향 데이터베이스 기술”, 정보과학회지, 11권, 2호, 1993, pp52
- [5] 권기현, “미래과학기술을 위한 대형연구시설 및 장비구축에 관한 정책연구”
- [6] 김승백 (2002), “주문생산을 위한 Job Shop 일정계획 시스템 개발”, 공학석사 학위논문.
- [7] 황규승 (1998), “생산관리”, 서울 홍문사.
- [8] Hitomi Katsundo(조규갑, 차성운 역), “생산시스템공학”, 사이텍미디어.
- [9] 송정수 (1998), “Job Shop 형태를 갖는 주문생산환경에서의 계층적 생산계획 및 통제 정보시스템”, 공학박사학위논문.
- [10] 강석호, 생산·운영관리, 경세원
- [11] 김원, “객체지향 데이터베이스 기술”, 정보과학지, Vol 11, No.2, 1993, 1995 pp.42-58

- [12] 박경수, 자재관리 및 재고통제, 구민사 1991
- [13] U Rembold, B.O. Nanji & A. Storr, Computer Intergrated Manufacturing and Engineering, Addison Wesley, 1993, pp 296
- [14] 회형군, 허계범 “객체지향소프트웨어공학”, 한국실리콘
- [15] 임중선, 주경수 “객체지향설계방법론을토대로한객체관계데이터베이스 시스템 설계및구현에관한연구”, 순천향대학교산업기술연구소
- [15] 조도형, 김종훈 “웹기반 입학 사정관리시스템설계및구현”, 순천향대학교정보기술공학부학사 학위논문
- [16] 강석호, 산업공학개론, 영지문화사, 1994, pp.317.
- [17] 조도형, 주경수 “관계형데이터베이스응용시스템을위한통합설계방법론개발-객체지향분석·설계방법론을 중심으로”, 한국컴퓨터정보학회논문집
- [18] 주경수 “관계 데이터베이스설계의 문제점 고찰” 순천향대학교논문집 15권1호
- [19] 임중선 “객체지향계방법론을토대로한객체관계데이터베이스시스템설계구현”, 순천향대학교석사학위논문