

스키마 변화에 따른 EPCIS의 효과 분석 및 검증에 관한 연구

이 중 석*

*남서울대학교 산업경영공학과

A Study on Analyzing the Effects and Verification of EPCIS by Influences of Shema

Zhong-Shi Li*

*Department of Industrial & Management Engineering, NAMSEOUL University

Abstract

RFID/USN, are recognized as the new growth engines for the future, regardless of the advanced and developing countries. RFID, in particular, already entered practical stage by global companies.

EPCIS, which is the one of the EPC global network components, makes a huge load on the system due to the large amount of entering data by following events.

In this study, a data model developed based on ER-WIN by collecting four kind information that occurred in the EPC global networks. This model supports both in processing of high performance and huge capacity of data by a considerable storage capacity and input speed. A simulation was developed in order to verify the performance. Each model tested several times and results were compared.

Keywords : RFID, Data Modeling, EPCIS, EPCglobal, Simulation.

1. 서론

최근 RFID 활용에 관한 국내외 성공사례가 속속 소개되면서 각 산업 주체들의 RFID 도입에 대한 관심이 높아지고 있다. 또한 계속되는 대내외적인 경기침체는 기업들로 하여금 그 어느 때보다도 생존과 번영을 위한 차별화 전략을 구사하도록 요구하고 있으며, 그 확실한 대안으로 RFID 도입이 떠오르고 있다.[10]

EPCglobal 네트워크에서는 이벤트가 빈번히 발생하고 이에 따라 입력되는 데이터 또한 대용량이며 이로

인하여 시스템에 엄청난 부하가 걸리게 된다.[6][1][8]

본 연구에서는 이를 해소하기 위하여 EPCglobal 네트워크에서 발생하는 네 가지 정보 데이터를 대상으로 ER-WIN에 기초하여 데이터 모델을 설계하고, Visual Basic 6.0으로 시뮬레이션을 구현하였으며, MS-SQL 2000으로 DB를 구축하였다. 시뮬레이션 결과 분석을 통하여 저장 용량과 입력 및 조회 속도를 모두 고려한 고성능 처리와 대용량 처리를 동시에 지원하는 데이터 모델에 대하여 효과 분석과 검증을 하고자 한다.

† 이 논문은 2012년도 남서울대학교 연구지원비에 의해 연구되었음.

† Corresponding Author: Zhong-Shi Li, Department of Industrial & Management Engineering, NAMSEOUL University., M·P : 010-2631-9780, E-mail: leezs0423@nsu.ac.kr

Received January 20, 2013; Revision Received March 5, 2013; Accepted March 16, 2013.

2. 이론적 배경

2.1 EPCglobal

EPCglobal은 상품 코드의 국제표준 개발/관리기구인 EAN과 UCC의 통합으로 탄생한 GS1이 2003년 11월에 설립한 자회사로서 EPC 코드와 EPCglobal 네트워크의 전 세계 보급을 총괄하고 있는 국제민간기구로 설립되었다. EPCglobal은 EPC 관리자 코드의 발급, 관리 및 ONS 등록 총괄, EPCglobal 네트워크 표준 개발 및 보급 등의 업무를 수행하고 있다. EPCglobal은 EPC의 표준화와 더불어 의약품 분야, 국방 분야, 화물, 세관, 항공 등에서 EPC 확산을 진행하고 있다.[9]

또한 EPCglobal 네트워크는 RFID 기술을 근간으로 물류정보 교환을 위해 물류 객체에 EPC를 할당하는 방법 및 이를 통한 기업 간의 정보 공유 방법에 대한 표준을 제공한다.[3][2] 이를 통해 EPC가 기록된 RFID 태그를 RFID 리더가 인식하여 그 정보가 EPCIS에 저장된다.[4]

2.2 EPCIS

EPCIS는 EPCglobal 네트워크 구성원 간의 데이터 교환의 주요 수단이다. EPCIS 데이터는 거래 파트너가 직접 관장할 수 없는 위치에 있는 개체로부터 발생하는 객체 또는 트랜잭션에 대한 데이터를 얻기 위해 공유하는 정보이다.[9] 이 EPCIS를 통해 EPCglobal 가입 자간 데이터 교환이 이루어진다.[5]

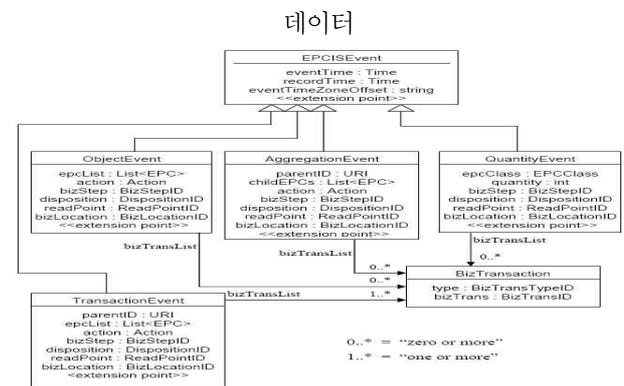
EPCIS는 정적 정보와 동적 정보를 가진다. 우선 정적 정보는 물리 객체의 고유한 성격에 대한 데이터로 상품명, 상품 코드와 같이 변경되지 않는 클래스 레벨의 정보와 제조일자, 유통기한과 같이 물리 객체마다 변경되는 인스턴스 레벨의 정보이다. 이를 통해 물리 객체의 동질성과 유일성을 제공한다.[8]

그에 반해 동적 정보는 물리 객체의 이동과 상태 변화에 따라 성장하고 변화한다.[8] EPCIS는 이벤트가 발생하면 ALE에서 전달해준 데이터로 해당 이벤트를 저장하는데, 상황에 따라 저장되는 데이터의 형태가 변경된다. EPCIS는 미들웨어로부터 태그 이벤트 정보를 받아 이를 이용해 상품의 상태 및 추적 정보를 생성하여 미래의 사용을 위해 로컬 저장소에 저장하고 관리한다.

2.3 EPCIS Event 데이터

EPCIS의 추상적인 데이터 모델은 크게 Event 데이터와 Master 데이터로 구성된다. Event 데이터는 비즈니스 프로세스를 거쳐 EPCIS Capture Interface를 통해 수집되어 향후 EPCIS Query Interfaces를 통해 사용되는 데이터라고 하면, Master 데이터는 Event 데이터를 설명하는 부가적인 정보를 담고 있는 데이터라고 할 수 있다[8].

Event 데이터는 다양한 공급망에서 발생하는 이벤트들을 표현하기 위해 한 개의 일반적인 이벤트와 4개의 세부 클래스로 구성된다.[9]



[Figure 1] EPCIS의 Event

<Table 1>은 EPCIS의 Event 데이터에 대하여 설명한 것이다.

<Table 1> EPCIS Event 데이터

구분	설명
EPCIS Event	모든 이벤트 타입에 대한 일반적인 클래스이다.
Object Event	EPC로 표시될 수 있는 사물에서 발생하는 이벤트로, 특정 시간(eventTime)에 특정 장소(bizLocation)에 특정 비즈니스 스텝(bizStep)에서의 EPC들의 List를 나타낸다.
Aggregation Event	물리적으로 합쳐진 사물을 EPC로 표시할 때 발생하는 이벤트로, 특정 시간(eventTime)에 특정 장소(bizLocation)에 특정 비즈니스 스텝(bizStep)의 parent EPC의 ParentID에 연결된 이벤트를 나타낸다.
Quantity Event	같은 EPC 클래스를 공유하게 되는 특정한 양의 사물에서 발생하는 이벤트로, 특정 시간(eventTime)에 특정 장소(bizLocation)에 특정 EPC 클래스에 속하는 제품의 수량을 나타낸다.
Transaction Event	EPC로 표시된 사물이 비즈니스 거래에 관련되었을 때의 이벤트이다.

2.4 데이터 모델링

데이터 모델링은 정보시스템을 구축하기 위해 해당 업무에 어떤 데이터가 존재하는지 또는 업무가 필요로 하는 정보는 무엇인지를 분석하는 방법이다.[12]

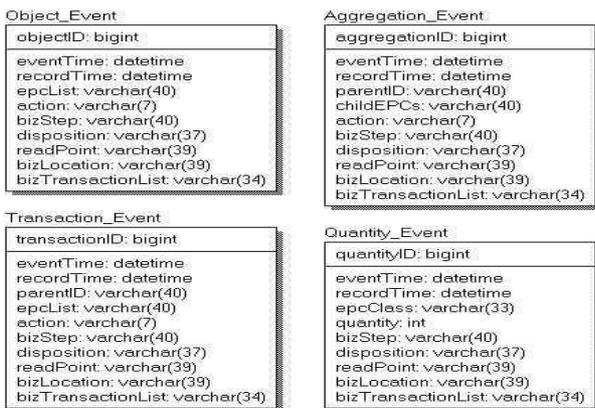
데이터 모델링은 단지 데이터베이스만을 구축하기 위한 용도로만 쓰이는 것이 아니라 데이터 모델링 자체로서 업무를 설명하고 분석하는 부분에도 매우 중요한 의미를 가지고 있다. 현실세계에서 데이터베이스까지 만들어지는 과정은 시간에 따라 진행되는 과정으로서 추상화 수준에 따라 개념적 데이터 모델, 논리적 데이터 모델, 물리적 데이터 모델로 정리할 수 있다.[11]

3. ER-WIN에 기초한 데이터 모델링

3.1 데이터 모델링 1

EPCglobal 네트워크에서 발생하는 데이터들은 크게 네 가지 정보 즉 Object Event Data, Aggregation Event Data, Quantity Event Data, Transaction Event Data로 구분된다.[6]

다음은 EPCIS 규격에 따른 테이블 구조를 바탕으로 작성한 데이터 모델이다.



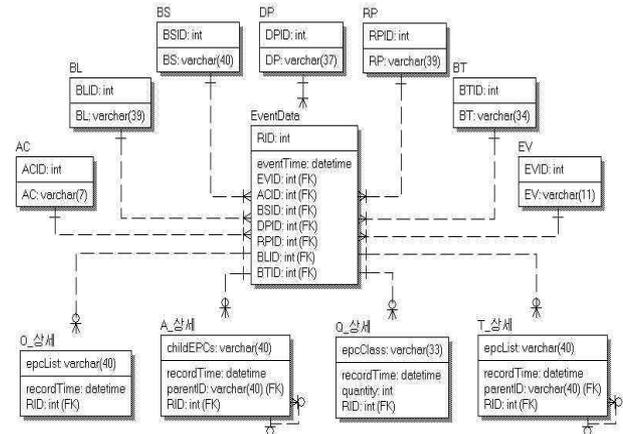
[Figure 2] 데이터 모델링 1

3.2 데이터 모델링 2

선행 연구[7]에서는 EPCIS 규격에 따라 정의된 각 EPCIS Event 데이터 구조를 기반으로 하여 저장되는 데이터는 많은 중복이 발생한다는 점과 이로 인하여 성능에 큰 영향을 미친다는 점을 지적하였으며, 이러한 데이터의 중복성 문제를 해결하기 위해 테이블을 분해하는 과정 즉 정규화 과정과 성능 향상을 고려한 역정

규화 과정을 거쳐야 한다고 제시하였다.

다음은 EPCIS 규격에 따라 정의된 테이블 구조를 정규화 및 역정규화 과정을 거쳐서 생성한 데이터 모델이다.

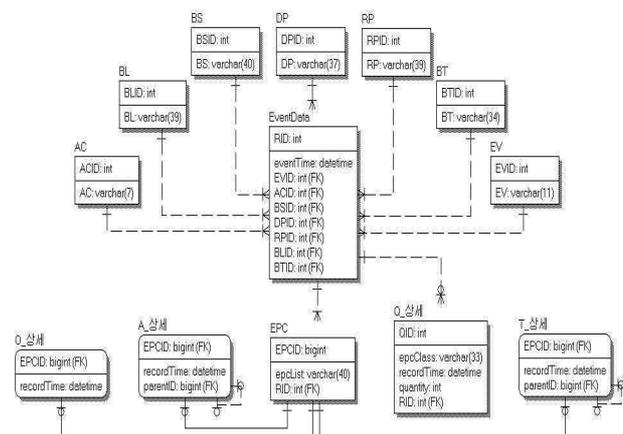


[Figure 3] 데이터 모델링 2

3.3 데이터 모델링 3

위의 데이터 모델 2는 비록 정규화 및 역정규화 과정을 거쳐 데이터의 중복을 많이 줄이고 또한 성능에도 어느 정도 향상을 가져올 수 있을 것이지만 4개의 상세 테이블에 중복되는 EPC 데이터들이 저장된다. EPCglobal 네트워크에서는 RFID 단말기로부터 수많은 EPC 데이터들이 지속적으로 발생한다. 따라서 이는 EPCIS에 엄청난 부하를 초래할 것이며 각 상세 테이블에 엄청난 데이터 중복을 초래할 것으로 고려된다.

다음은 4개의 상세 테이블에 중복으로 저장되는 EPC 데이터들을 따로 분리하여 새로운 테이블을 생성하여 저장하고, 각 상세 테이블에서는 EPC 데이터들의 ID를 저장하는 구조로 변경한 데이터 모델이다.

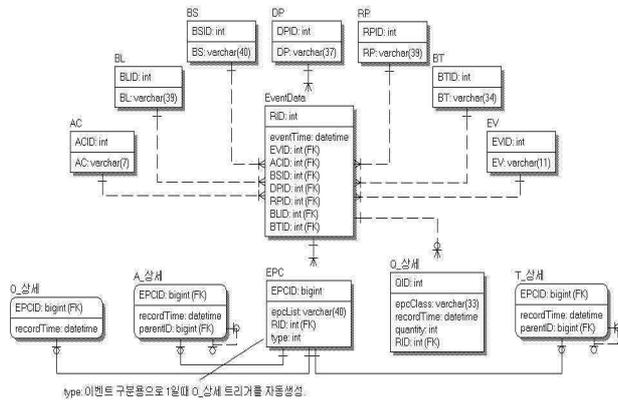


[Figure 4] 데이터 모델링 3

3.4 데이터 모델링 4

위의 데이터 모델 3은 대용량으로 저장되는 EPC 데이터를 분리하여 따로 저장하여 중복을 줄임으로써 성능 향상에 기여하였다. 다만 테이블 분리로 인하여 삽입 프로세스가 기존보다 많이 발생하게 될 것이며 이는 곧 데이터 저장에 많은 시간을 소요한다는 것이다. 즉 데이터 모델 3에서는 데이터 모델 2에 비해 저장 용량은 많이 줄일 수 있는 반면 삽입 프로세스가 증가함에 따라 저장 시간이 오히려 더 많이 걸린다는 것이다.

다음은 삽입 프로세스의 증가로 인한 저장 시간을 단축하기 위하여 업무 성격상 4개의 이벤트에서 거의 반복으로 발생하는 Object 이벤트에 대하여 삽입 트리를 추가한 데이터 모델이다.



[Figure 5] 데이터 모델링 4

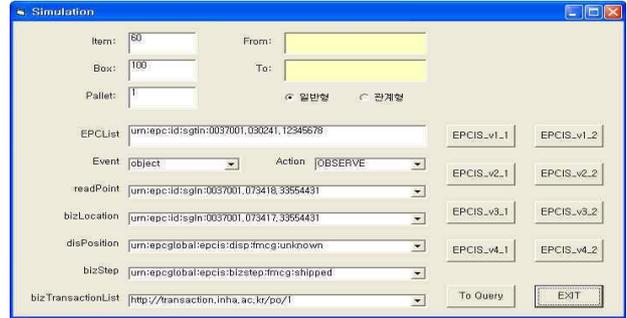
4. 시뮬레이션

4.1 시뮬레이션 설계 및 구현

각 데이터 모델에 대하여 효과 분석 및 검증을 하기 위하여 다음과 같이 시뮬레이션을 구현하고, 각 모델별로 동일한 양의 데이터를 저장할 때 소요되는 입력 시간과 디스크의 저장 용량을 비교하고, 동일 데이터를 검색하는데 소요되는 조회 시간을 비교하고자 한다.

먼저 시뮬레이션 인터페이스이다. 화면 상단은 좌측의 item, box, pallet의 수에 대한 입력 부분과, 우측의 시뮬레이션의 시작과 종료시간 및 데이터베이스 종류를 구분하는 옵션으로 구성된다.

화면 하단은 좌측의 실제 데이터베이스에 저장되는 이벤트 데이터의 필드 값들과, 우측의 각 모델에 대한 다양한 실행 버튼 8개와 쿼리 인터페이스로 이동하기 위한 ToQuery 버튼, 화면 종료용 EXIT 버튼으로 구성된다.

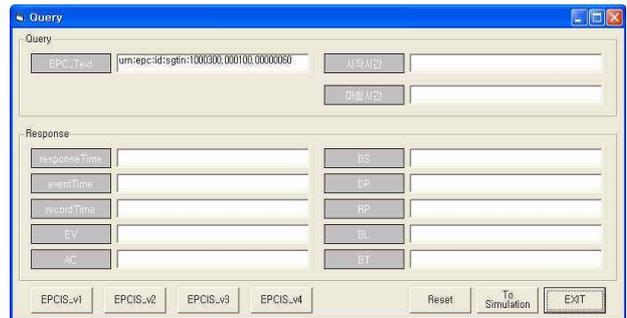


[Figure 6] 시뮬레이션 인터페이스

다음은 쿼리 인터페이스이다. 화면 상단은 검색 대상인 EPC_Text를 입력하는 부분과 검색에 사용되는 시간을 계산하기 위한 시작시간과 마침시간 표시하는 부분이다.

화면 가운데는 결과 표시 부분으로서 결과를 반환하는 responseTime을 포함하여 eventTime, recordTime, event, action, bizStep, disPosition, readPoint, bizLocation, bizTransactionList 등 이벤트가 발생할 때 함께 저장되었던 정보들을 보여준다.

화면 하단은 버튼 구역으로서 각 모델들로부터 데이터를 검색하기 하기 위한 검색용 버튼 4개와 화면 데이터를 초기화하는 Reset 버튼, 시뮬레이션 인터페이스로 이동하기 위한 ToSimulation 버튼, 화면 종료용 EXIT 버튼이 있다.



[Figure 7] 쿼리 인터페이스

4.2 시뮬레이션 결과 및 분석

위에서 제시한 데이터 모델들을 이용하여 시뮬레이션 프로그램을 실행시키고 각 모델에 대하여 여러 번 테스트를 거쳐 그 결과를 비교해 보았다.

또한 발생하는 이벤트에 대하여 Object 이벤트만 발생하는 경우와 모든 이벤트가 동시에 발생하는 경우를 가정하여 테스트를 진행하였다.

먼저 Object 이벤트 횟수에 따른 시뮬레이션의 입력 시간에 대한 수행결과는 다음과 같다.

<Table 2> Object 이벤트 입력 시간 비교

[단위 : 초]

횟수	모델 1	모델 2	모델 3	모델 4
1	4	4	8	6
2	8	6	15	9
3	14	10	20	12
4	18	14	26	16
5	23	16	34	21
6	27	20	39	25
7	31	22	49	28
8	33	25	54	33
9	43	28	60	37
10	42	31	66	44
15	63	53	98	59
20	79	71	126	77
25	96	92	174	94
30	114	105	196	115
35	132	122	215	133
40	158	138	263	187
45	171	157	280	190
50	187	172	315	192
100	369	340	627	368
200	732	675	1,236	751
300	1,086	1,023	1,842	1,066

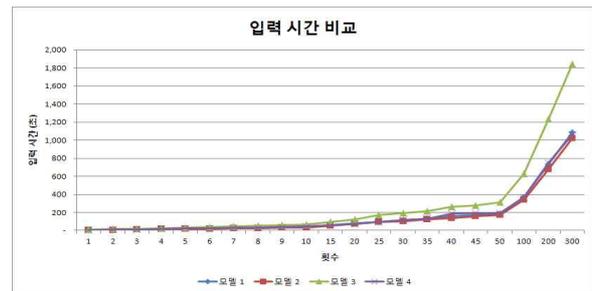
Object 이벤트 횟수에 따른 시뮬레이션의 저장 용량에 대한 수행결과는 다음과 같다.

<Table 3> Object 이벤트 저장 용량 비교

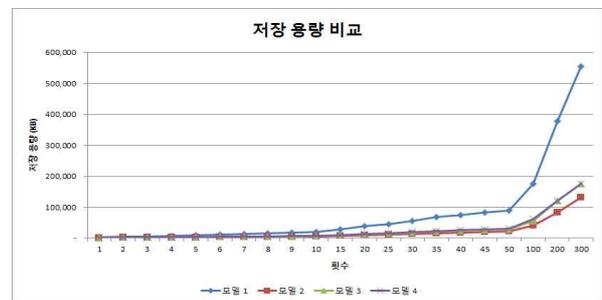
[단위 : KB]

횟수	모델 1	모델 2	모델 3	모델 4
1	2,688	1,600	1,792	1,792
2	4,544	2,304	2,432	2,432
3	6,080	2,432	3,008	3,008
4	8,192	3,008	3,328	3,712
5	9,920	3,328	4,096	4,096
6	12,032	3,712	4,544	4,992
7	13,248	4,096	4,992	5,504
8	16,064	4,544	6,080	6,080
9	17,664	4,992	6,080	6,720
10	19,456	5,504	6,720	7,424
15	28,608	7,424	9,920	9,920
20	38,208	9,920	13,248	13,248
25	46,272	12,032	16,064	16,064
30	56,064	13,248	17,664	19,456
35	67,904	16,064	21,440	21,440
40	74,688	17,664	23,616	25,984
45	82,176	19,456	25,984	28,608
50	90,432	21,440	28,608	31,448
100	176,448	42,048	56,064	61,696
200	378,496	82,176	120,448	120,448
300	554,304	132,544	176,448	176,448

<Table 2>와 <Table 3>의 결과를 차트로 표현하면 다음과 같다.



[Figure 8] Object 이벤트 입력 시간 비교



[Figure 9] Object 이벤트 저장 용량 비교

위의 결과에서 알 수 있듯이 Object 이벤트만 발생하였을 때 입력 시간 비교에서는 모델 3, 모델 1, 모델 4, 모델 2 순으로 모델 3이 입력 시간이 제일 많이 소요되었고 모델 2가 제일 적게 소요되었다.

모델 3과 모델 4는 관계형 모델로서 EPC 테이블과 O_상세 테이블에 동시에 삽입 프로세스가 발생한다. 단 모델 4는 Object 이벤트에 대하여 삽입 트리거를 추가한 모델로서 모델 3에 비해 입력 시간이 단축된다. 모델 2 역시 관계형 모델이지만 O_상세 테이블에만 삽입 프로세스가 발생하기 때문에 모델 3과 모델 4에 비해 입력 시간이 적게 소요된다. 또한 모델 1보다 테이블의 인스턴스의 길이가 짧기 때문에 모델 1보다도 입력 시간이 적게 소요된다.

반면 저장 용량 비교에서는 모델 1, 모델 4, 모델 3, 모델 2 순으로 모델 1이 디스크 용량을 제일 많이 차지하였고 모델 2가 디스크 용량을 제일 적게 차지하였다.

모델 2와 모델 3 및 모델 4는 관계형 모델로서 중복이 발생하는 필드를 테이블로 분리하였기 때문에 많은 중복을 줄였고 이로 인하여 모델 1에 비해 디스크 용량을 크게 줄일 수 있다.

모델 4는 모델 3에 비해 Object 이벤트에 대하여 삽입 트리거를 추가한 모델로서 디스크 용량에는 별로 차이가 발생하지 않는다.

다음으로 모든 이벤트 횟수에 따른 시뮬레이션의 입력 시간에 대한 수행결과는 다음과 같다.

<Table 4> 모든 이벤트 입력 시간 비교

[단위 : 초]

횟수	모델 1	모델 2	모델 3	모델 4
1	10	9	11	9
2	19	16	22	17
3	27	23	30	24
4	34	31	40	30
5	41	38	52	37
6	50	45	58	44
7	58	51	67	52
8	66	60	78	60
9	74	68	90	65
10	81	73	98	73
15	120	107	141	111
20	163	140	200	141
25	206	179	236	178
30	230	211	287	212
35	262	249	333	246
40	310	285	374	280
45	339	313	432	315
50	416	351	484	359
100	733	713	964	694
200	1,483	1,420	1,915	1,386
300	2,255	2,035	2,815	2,067

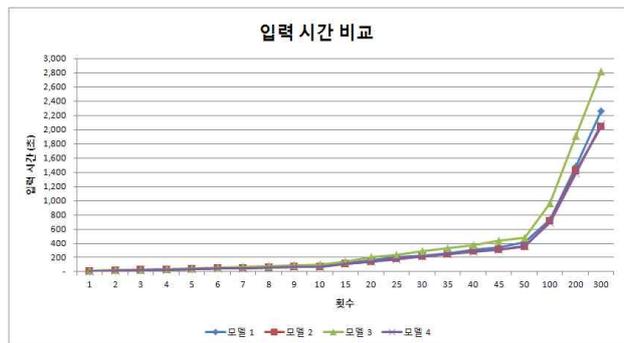
모든 이벤트 횟수에 따른 시뮬레이션의 저장 용량에 대한 수행결과는 다음과 같다.

<Table 5> 모든 이벤트 저장 용량 비교

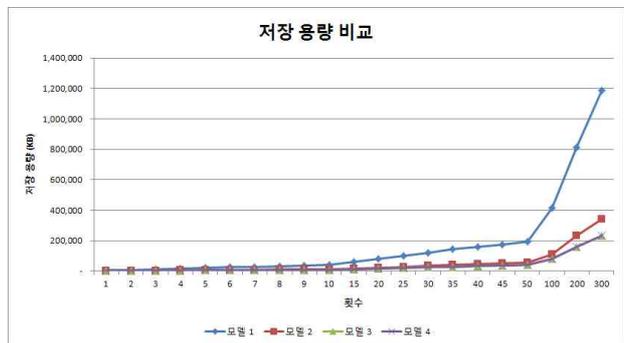
[단위 : KB]

횟수	모델 1	모델 2	모델 3	모델 4
1	4,992	2,432	2,176	2,176
2	9,024	3,328	3,008	3,008
3	13,248	4,544	3,712	3,712
4	17,664	5,504	4,544	4,544
5	21,440	6,720	4,992	5,504
6	25,984	8,192	6,080	6,080
7	28,608	9,024	6,720	6,720
8	31,488	9,920	7,424	8,192
9	38,208	10,944	8,192	8,192
10	42,048	12,032	9,024	9,024
15	61,696	17,664	13,248	13,248
20	82,176	23,616	17,664	17,664
25	99,520	28,608	21,440	21,440
30	120,448	34,688	25,984	25,984
35	145,792	42,048	28,608	28,608
40	160,384	46,272	31,488	34,688
45	176,448	50,944	38,208	38,208
50	194,112	56,064	42,048	42,048
100	416,384	109,504	82,176	82,176
200	811,584	234,944	160,384	160,384
300	1,188,288	344,064	234,944	234,944

<Table 4>와 <Table 5>의 결과를 차트로 표현하면 다음과 같다.



[Figure 10] 모든 이벤트 입력 시간 비교



[Figure 11] 모든 이벤트 저장 용량 비교

위의 결과에서 알 수 있듯이 모든 이벤트가 동시에 발생하였을 때 입력 시간 비교에서는 모델 3, 모델 1, 모델 4, 모델 2 순으로 모델 3이 시간이 제일 많이 소요되었고 모델 2가 제일 적게 소요되었다.

이는 Object 이벤트만 발생하였을 때와 같은 방식으로 해석되며 결과도 동일하다.

반면 저장 용량 비교에서는 모델 1, 모델 2, 모델 4, 모델 3 순으로 모델 1이 디스크 용량을 제일 많이 차지하였고 모델 3과 모델 4가 디스크 용량을 제일 적게 차지하였다.

모델 2와 모델 3 및 모델 4는 관계형 모델로서 중복이 발생하는 필드를 테이블로 분리하였기 때문에 많은 중복을 줄였고 이로 인하여 모델 1에 비해 디스크 용량을 크게 줄일 수 있다.

모델 3과 모델 4는 모델 2에 비하여 중복되는 EPC 데이터들을 EPC 테이블로 분리하였기 때문에 중복이 줄어들어 디스크 용량을 많이 줄일 수 있다.

모델 4는 모델 3에 비해 Object 이벤트에 대하여 삽입 트리거를 추가한 모델로서 디스크 용량에는 별로 차이가 발생하지 않는다.

마지막으로 시물레이션 결과에 따른 데이터 중에서 원하는 EPC 정보를 검색하기 위하여 쿼리 프로그램을 실행시키고 각 모델에 대하여 동일한 EPC 데이터를 검색하는 테스트를 여러 번 수행하여 그 결과를 비교해 보았다.

각 모델들은 데이터의 저장 용량을 줄이는 면에서 많은 성능 향상을 보이지만 테이블이 많이 분리되다 보니 조회 성능에는 많은 악영향을 미칠 것이다. 즉 필요한 EPC 데이터를 검색하기 위하여 여러 번의 조인을 거쳐야만 원하는 정보를 얻을 수 있다. 따라서 보다 빠르게 데이터를 검색하기 위하여 각 모델에 뷰를 추가하였다.

또한 테스트에 사용된 데이터는 위의 시물레이션 프로그램 횟수를 300(EPC 수량 1,830,300개), 500(EPC 수량 3,050,500개), 1000(EPC 수량 6,101,000개)으로 하였을 때 저장된 데이터를 기준으로 하였으며 각 모델마다 맨 마지막으로 저장된 데이터를 검색하였다.

이벤트 횟수에 따른 저장 용량 및 조회 시간에 대한 수행결과는 다음과 같다.

<Table 6> 저장 용량 비교

[단위 : KB]

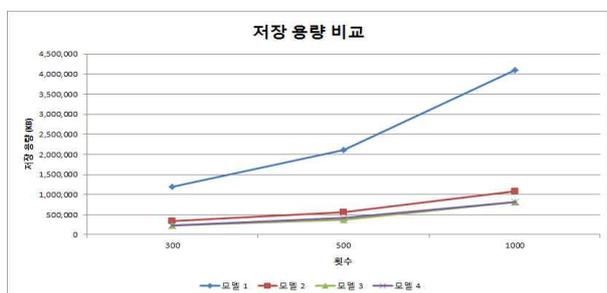
횟수	모델 1	모델 2	모델 3	모델 4
300	1,188,288	344,064	234,944	234,944
500	2,105,344	554,304	378,496	416,384
1000	4,102,976	1,080,256	811,584	811,584

<Table 7> 조회 시간 비교

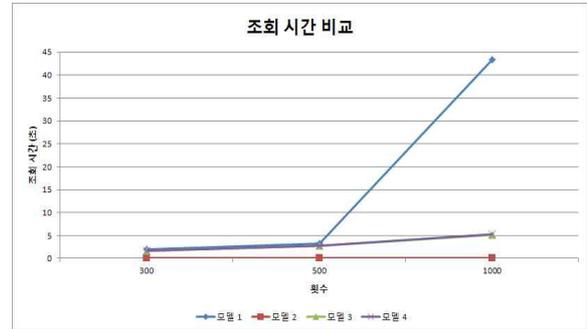
[단위 : 초]

횟수	모델 1	모델 2	모델 3	모델 4
300	2.0	0.0	1.6	1.6
500	3.2	0.1	2.7	2.8
1000	43.4	0.2	5.2	5.3

<Table 6>과 <Table 7>의 결과를 차트로 표현하면 다음과 같다.



[Figure 12] 저장 용량 비교



[Figure 13] 조회 시간 비교

위의 결과에서 알 수 있듯이 조회 시간 비교에서는 모델 1, 모델 4, 모델 3, 모델 2 순으로 모델 1이 시간이 제일 많이 소요되었고 모델 2가 제일 적게 소요되었다.

모델 1은 단일 테이블에 모든 데이터들이 저장된 경우로 검색 시 테이블 스캔과 같은 효과가 나타나며 이로 인하여 나중에 저장된 데이터일수록 조회 시간이 많이 소요된다.

모델 2와 모델 3 및 모델 4는 관계형 모델로서 많은 조인이 발생함에도 불구하고 뷰를 통하여 조회 시간을 많이 단축할 수 있다. 이들 중 모델 2는 모델 3과 모델 4에 비해 EPC 데이터에 대한 조인이 없기 때문에 결과를 얻는데 시간이 더욱 단축된다. 또한 모델 4는 모델 3에 비해 Object 이벤트에 대하여 삽입 트리거를 추가한 모델로서 저장 용량 및 조회 시간에는 별로 차이가 발생하지 않는다.

즉, 모델 2, 모델 3, 모델 4는 모두 모델 1에 비해 우수하며 이로서 모델 1의 단점인 입력 시간, 저장 용량, 조회 시간을 극복하였다고 볼 수 있다. 또한 모델 3과 모델 4는 저장 용량과 조회 시간에서는 극히 미세하지만 입력 시간에서는 엄청 차이나 발생하므로 모델 4가 모델 3에 비해 좀 더 우수하다. 모델 2와 모델 4는 입력 시간과 조회 시간에서 역시 극히 미세하지만 저장 용량에서는 엄청 차이가 발생하므로 모델 4가 모델 2에 비해 우수하다. 이를 순위별로 정리하면 다음과 같으며, 모델 4가 입력 시간, 저장 용량, 조회 시간을 종합적으로 보았을 경우 가장 우수한 모델이라는 것을 알 수 있다.

<Table 8> 순위 비교

구분	모델 1	모델 2	모델 3	모델 4
입력 시간	3	1	4	2(1)
저장 용량	4	3	1	2(1)
조회 시간	4	1	2	3(2)

5. 결 론

우리가 흔히 얘기하는 성능에는 크게 두 가지 종류가 있는데 조회 성능과 삽입 성능이다. 빠르게 추출해야 하는 조회 성능과 빠르게 저장해야 하는 삽입 성능은 다르므로 막연하게 성능이라고 표현하면 의미를 정확하게 전달하지 못할 것이다. 이 두 가지 성능이 모두 우수하면 좋겠지만 데이터 모델을 구성하는 방식에 따라 두 성능이 Trade-Off 되어 나타나는 경우가 많이 있다.[11][13]

본 연구에서는 EPCglobal 네트워크에서 발생하는 네 가지 정보 데이터를 파악하여 ER-WIN에 기초하여 데이터 모델을 설계하였다. 작성된 모델은 저장 용량과 입력 및 조회 속도를 모두 고려하여 고성능 처리와 대용량 처리를 동시에 지원한다. 이를 검증하기 위하여 시뮬레이션을 개발하여 각 모델에 대하여 여러 번 테스트를 거쳐 그 결과를 비교해 보았다.

추후 연구과제로는 데이터 모델의 확장과 현업 프로세스를 기반으로 하는 시뮬레이션 효과 분석 및 검증에 대한 연구가 필요할 것이다.

6. 참 고 문 헌

- [1] Chang-Ho Lee, "A Study on EPCIS Repository and Hybrid DBMS Optimization for Efficient EPC Network", INHA Industry Partnership Institute, 2010.
- [2] Donggyu Jung, Jaekwan Park, Bonghee Hong, "Distributed Continuous Query Processing in EPCglobal Network", Proceedings of the KIISE, Vol.35, No.2(C), 2008.
- [3] Honggoo Moon, Gideok Han, Hyuckchul Kwon, "Method for Logistics Information Access Control on EPCglobal Network through the Interaction between EPC IS and EPCIS Discovery System", Proceedings of the KIISE, Vol.34, No.2(D), 2007.
- [4] Jiho Kim, Giwon Shin, Daewon Park, Hyuck-Chul

Kwon, "Expansion of EPCIS and EPCDS for Efficient Distributed Logistics Information Management", Journal of KIISE, Vol.16, No.11, 2010.

- [5] Jinseung Kim, "Efficient very large data processing for EPCIS using stream data manager", KYUNGHEE UNIVERSITY, Master's Thesis, 2007.
- [6] Zhong-Shi Li, Chang-Ho Lee, "A Study on the Efficiency of the EPCIS using Simulation", Journal of the Korea safety Management and Science, Vol.12, No.4, 2010.
- [7] Zhong-Shi Li, "A Study on EPCIS system Modeling by Data Modeling Method", Journal of the Korea safety Management and Science, Vol.14, No.2, 2012.
- [8] Yong-Chul Jho, "An Efficient EPCIS Repository for RFID based Integrated Logistics Center", INHA UNIVERSITY, Doctoral Dissertation, 2009.
- [9] RFID-GL 기술자격검정, 영진미디어.
- [10] SCM에서 RFID 활용하기, 한국경제신문.
- [11] SQL 전문가 가이드, 한국데이터베이스진흥원.
- [12] SQL Server 2005를 위한 데이터베이스 모델링, dbBADA, 김연홍.
- [13] <http://www.dbguide.net>

저 자 소 개

이 종 석



인하대학교 산업공학과 공학박사 취득. 현재 남서울대학교 산업경영공학과 교수로 재직 중.
관심분야: EPCglobal Network, 시뮬레이션, RFID를 활용한 응용시스템, SCM, DB 등.

주소: 충남 천안시 성환읍 매주리 21 남서울대학교 산업경영공학과