

시나리오 역할 게임을 통한 신속 시스템 설계/ 검증 방법

- 패키지 응용 시스템을 대상으로 -

예도경¹⁾ · 이용재¹⁾ · 강두호²⁾ · 최열현¹⁾

¹⁾LG-EDS 시스템 CALS & CIM 연구소

²⁾LG 전자 생산기술센터 주임연구원

Rapid System Design/Validation Method through Scenario Based Role Playing Game - for Package Based System Implementation -

Do-Kyung Ye, Yong-Jae Lee, Doo-Ho Kang and Yul-Hyun Choi

ABSTRACT

Recently, many enterprises wish to preoccupy competitive positions in the market by motivating themselves towards CALS(Commerce At Light Speed) and try hard to achieve system implementation with the use of new concept-based backbone systems such as ERP(Enterprise Resource Planning) or PDM(Product Data Management). However, unfortunately there is an inevitable discrepancy during implementation period between package vendors and customers due to lack of clear methodology and misunderstanding about system. In some cases, consensus on partially verified system design invokes customers' continuous change demands and often it tends to result in unnecessary project delays. PDM application system implementation project of the company L has been studied here to solve these sorts of problems. The major yields of this work are achievement of a willing approval from customers and rapid system design under a time-limited and conceptually defined project scope and finally, through fully verified system design, we were able to reach successful system implementation.

1. 서 론

급변하는 시장환경과 이에 대한 대응을 위한 여러 기업들의 노력은 CALS 환경에서의 생존을 위한 여러 가지 형태의 시스템 구현 노력으로 연계되어 진행되고 있다. ERP나 PDM을 중심으로 대부분은 이미 상용 제품화되어 있는 패키지를 도입하여 많이 구현되고 있으며, 이는 신속한 시스템의 구현 및 검증된 안정성의 확보, 신기술의 지속적인 업그레이드 지원 등등 많은 장점을 고려한 것이라 할 수 있다.

패키지를 통한 구현에 있어서도 많은 경우 자사의

환경에 맞는 커스터마이징이 발생하게 되는데, 대부분의 경우 기존의 개발방식에 근거한 설계 및 접근 방식으로 불필요한 작업에 많은 시간을 허비하고 있어 패키지의 장점을 살리기 보다는 역효과를 초래하기 쉽다. 또한, 대부분의 이들 패키지들은 신개념에 근거한 것들로서 패키지를 제공하는 벤더 혹은 개발 측의 관점과 고객들의 이해수준에 많은 차이가 발생하므로, 커스터마이징 이전에 시스템의 이미지 공유가 더 우선해야 한다.

본 연구에서는 이상과 같은 문제점들에 대한 고려와 DFx(Design For x...)의 개념, 그리고 시뮬레이터

드 어널링 등의 개념들을 시스템 개발에 고려한 것으로서, David A. Taylor가 제시한 Convergent Engineering 개념을 시스템 분석/설계를 응용해 보았다²⁾. 상용 PDM 시스템인 IMAN을 근간으로 한 응용시스템 개발에 직접 이 방법을 도입하였으며, 3주만에 설계서에 대해 개발측과 고객측과의 합의를 도출해 낼 수 있었다. 현재 변경사항 최소화 상태에서 원래 계획대로 차질 없이 개발이 완료되어 현업 안정화 단계에 있다.

본 연구에서의 핵심 되는 두 가지 요소는 '시나리오'와 'GUI Object Class Card'이며, 전자를 위해서 개발될 시스템의 사용자 동작 시나리오를 IDEF3 형태를 응용하여 정의하였고, 각각에 해당되는 GUI 화면들을 각각의 Action과 Attribute 별로 나누고 제반 제약사항들을 정의한 GUI Object Class Card로 만들어 역할 게임의 도구로 사용하였다. 3차례의 역할 게임을 통한 검증 이후 이에 대해 파워포인트 형태의 GUI로 구성된 시나리오 근간 데모를 통해 전체적 공유 및 합의를 이끌어 내는데 성공하였다. 그리고, 이를 통한 안정적인 패키지 응용 시스템 구현을 유도할 수 있었다.

2. L 사의 PDM 패키지 응용 프로젝트 개요

L 사는 96년도부터 전사적으로 PDM 도입하기 위한 TFT를 구성하고, 마스터플랜 및 표준 패키지를 EDS/UG의 IMAN(PDM 패키지)으로 선정하여 1차 '기술정보 관리시스템' 구축에 이어 현재 2차 시스템인 '설계정보 관리시스템'의 안정화를 실시하고 있다. L 사는 PDM 추진을 위하여 아래 그림에서 보는 바와 같은 T-technique 적 시스템 접근방법을 바탕으로 PDM 시스템으로의 구현을 접근하고 있으며, 이

는 넓이우선(bread-first) 접근을 위한 TFT의 마스터플랜 추진과 깊이우선(depth-first) 접근을 위한 Pilot 시스템의 구현, 그리고 Pilot 시스템에서 얻은 시행착오의 노하우를 바탕으로 단계적 시스템화를 피하는 방법의 반복에 의한 것이다³⁾.

PDM의 단계적 시스템화도 정보관리를 위한 시스템인 만큼 제품정보의 3가지 카테고리를 기준으로 추진해 나가고 있으며, 프로세스 및 업무 문화적으로 가장 위험요소가 적은 기술정보를 중심으로 1차 Pilot 시스템을 구현하고, 여기서 얻은 시행착오를 바탕으로 가장 핵심이 되는 제품정의정보에 대한 2차 시스템을 구현, 최종적으로 관리/운영정보에 까지 시스템 구현을 통해 PDM의 전체 모습으로 접근해 나가고 있다.

각 단계별 시스템을 구성하는데 있어서도 Pilot에서의 Feedback과 전체 PDM의 구성을 고려한 초기 마스터플랜의 지속적인 재검토를 실시하였으며, 이에 개발 벤더와 고객은 지속적인 인터뷰를 통해 CSE(Concurrent Simultaneous Engineering) 개념에 입각한 4가지 요소 - 즉, 조직(Organization), 프로세스(Process), 자원(Resource), 제품(Product)를 기준으로 3가지 방향 - 즉, 병행화(Parallelization), 표준화(Standardization), 통합화(Integration)로의 개선점을 도출해 나가려 하였다⁴⁾. 즉, Role-Based Organization Chart의 구성 및 IDEF0 모델링을 통한 업무프로세스의 명확한 파악, 그리고 제품과 관련된 모든 정보의 검토를 분석해 나갔으며, 고객 요구와 PDM 시스템적인 지원 기능들을 대응시켜 추진할 시스템의 개략적 모듈 범위까지를 도출해 내고, 도출된 시스템의 범위를 구현하는 프로젝트를 상호합의 하에 착수하여 본 연구에서 제시하는 방식을 통해 시스템의 설계를 진행



그림 1. L 사의 단계적 PDM 시스템화 방안

하였다^{1,4,5)}.

3. 시나리오 역할 게임을 통한 신속 시스템 설계/검증 방법

본 연구에서 제시하는 방법은 객체지향적 접근을 통한 패키지 시스템 구현을 위한 설계/검증이 가능하도록 제시되었고, 신속한 설계/검증을 통한 고객-개발 벤더간의 합의 도출을 위해 결과 지향적 산출물의 양식으로도 제시될 수 있도록 추진되었다.

◎ IDEF3를 응용한 시나리오의 정의

IDEF3는 시나리오 지향적 프로세스 흐름 모델화 방법으로서, 특정 시스템이나 조직이 작동하는 방식에 관하여 유력한 전문가의 지식을 표현하는 구조적 방법을 제공하는 것을 목적으로 한다^{6,7)}. IDEF3의 목적에 대해 제시된 여러가지 목적 중 "시스템 디자인과 디자인 트레이드 오프 분석을 지원한다"의 항목을 고려하여 IDEF3 방법의 응용을 본 연구의 방법에 고려하게 되었으며, 이미 IDEF0를 통해 어느 정도 업무 프로세스가 파악된 상태에서 사용자의 '행동적' 측면을 고려한 시스템 설계에는 가장 적합한 방법이라 판단하고 적용하였다⁷⁾.

우선적으로 시나리오 리스트를 작성하고, 각각의 시나리오별로 IDEF3를 이용한 사용자의 시스템 운영을 고려한 동작 시나리오를 구성하였다. 기본적인 표현방식은 IDEF3 그대로를 따랐으나, IDEF Ref. # 자리에 GUI Object Class Card의 번호와 그 Attribute 혹은 Action을 표현하여 시나리오에 따른 시스템의 실제 구동상황을 연계하여 검증될 수 있도록 하였다. 신속한 시스템 설계의 개발 벤더와 고객간의 합의를 주된 목적으로 하였으므로 OSTN(Object Status Transaction Network)까지는 정의하지 않았으며, 사용자 동작 시나리오 정도의 역할 게임이 가능한 수준까지만을 정의하였다(그림 2. 참조).

◎ GUI Object Class Card의 작성을 통한 화면 설계

GUI Object Class Card는 최초 수작업을 통하여 Sketch 해 나갔으며, Attribute 및 Action의 정의를 동시에 수행하였다. 참고로 이 때 상호간 용어에 대한 정의와 여러 상황에서의 제약 조건들을 동시에 검토해 나갔다. 물론, 패키지를 최대한 그대로 활용하는 범위에서의 접근을 시도하였으므로 기본 기능 이외의 추가적인 주요 화면들로 구성하였다. GUI를 각각 Object Class로 간주하고 Class 정의 정보들을 형태별

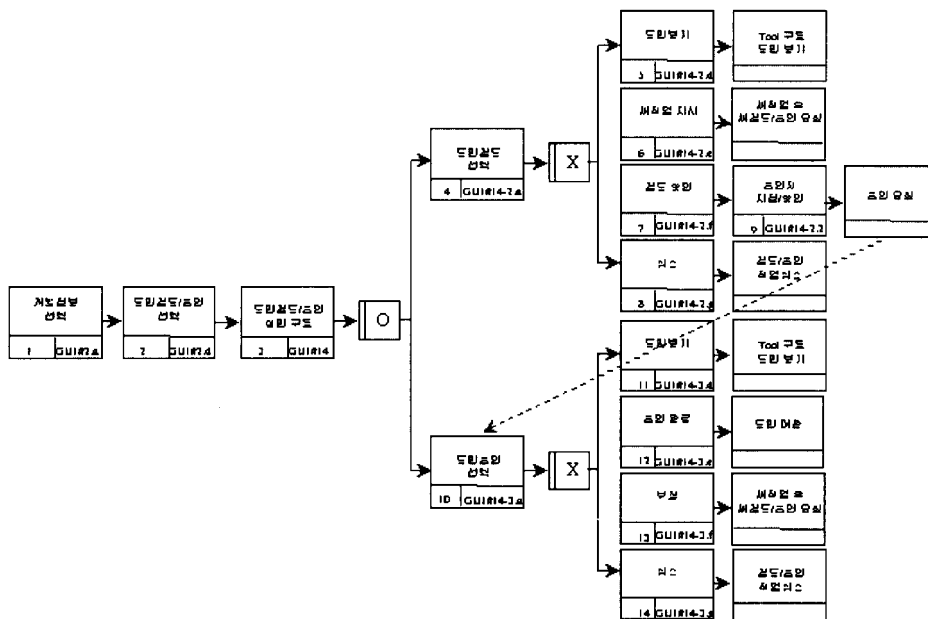


그림 2. IDEF3를 응용한 시나리오 예제 : 검토/승인 시나리오

5.6	저 목	화면 설계	작 성 자	이동재 / 예도경	작 성 일	1997.12.9
	작성 목적	사용자와의 interface를 위한 각 화면구성을 정의		비 전	3.0	
				페이지	21 / 24	
GUI#14.2 도면 검토/승인 화면(도면검토)						
		C/A/Part	타입	설명		
Components	①	TextBox	검도일자를 자동 Display			
	②	ComboBox	승인자가 자동 Display되고, 검도자가 승인자를 다시 지정할 수 있다			
	③	Grid	상태가 "검도외회"인 도면파일 정보를 Class에서 정보를 가져옴			
	④	Grid	상태가 "검도완료"인 도면파일 정보를 Class에서 정보를 가져옴			
	⑤	Status Bar	현재 Action에 대한 상태를 표시			
Actions	⑥	TopCell	도면검토 화면 Display			
	⑦	Cell	3의 모든 Row를 선택			
	⑧	Button	특정 도면 항목의 Row를 선택			
	⑨	Button	특정 검도 항목 도면을 읽는다			
	⑩	Button	3에서 선택한 도면을 제작업 지시부여			
	⑪	Button	선택된 도면에 대한 검도완료 부여			
	⑫	Button	검도/승인 화면 Close			

그림 3. GUI Object class Card 예제 : 검도/승인 GUI

로 각각 다르게 하여 Action의 경우 알파벳으로, Attribute의 경우 숫자로 Naming 하였다. 또한, 각 GUI Object class Card 마다 번호를 매겨 IDEF3를 통하여 정의하는 시나리오와의 연계검증이 가능하도록 하였다. 예를 들어, IDEF3로 정의된 시나리오의 IDEF Ref #이 GUI#14.2.a 일 경우는 GUI Object Class Card 번호가 '14.2'인 화면에서의 'a'로 정의된 Action을 수행한 경우로 연계검토 하도록 한 것이다(그림 3. 참조).

패키지에서 커스터마이징을 위한 화면설계가 완료되었다는 것은 시스템의 구현이미지가 거의 다 도출되었다 라고 말해도 과언이 아니며, 따라서, 가장 핵심적인 설계이슈로서 지속적인 고객과의 검토를 통한 변경이 가능하도록 구성/활용하였다.

◎ 시나리오 역할 게임의 실시와 GUI 프로토타입의 구성

시나리오의 작성 및 GUI Object Class Card는 앞서 수행하였던 현상분석자료와 요구사항 수렴을 위한 인터뷰에서 각각 그 초안이 작성되며, 이 도구들을 가지고 실제로 프로젝트 추진 팀원들간에 Validating Model을 통해 진행하였다. 즉, 'Talk Through' - 'Walk Through' - 'Run Through'의 3단계를 통해 구체화 시켜나갈 수 있었으며, 이러한 작업도중에 각

상황마다의 이슈 사항들을 정리하여 제약 사항들에 대해서도 명확히 할 수 있었다²⁾. 우선, Talk Through 단계에서는 시나리오 초안과 GUI 초안을 두고 각각에 대해 검토하는 과정으로서 실제 속성들의 화면 내 배치까지도 정의해 나가는 형태로 추진하였다. 단, 자칫 잊어 버리기 쉬운, 시스템 구현 목적을 상호 계속적으로 상기시켜 방향성을 잃지 않도록 하였다. 가장 불확실할 수 있는 이 단계는 제품설계에서의 Concept Design과 같이 가장 중요한 부분이며, 따라서 시스템의 구현 목적과 효과를 추진팀원 모두가 확실히 공유하여 방향성을 잃지 않도록 하는 것이 중요하다.

Walk Through 단계에서는 1차 검토되었던 것을 반영한 시나리오와 GUI를 가지고 역할 게임을 실시하여 미처 반영하지 못했던 사항이나 문제점 등에 대해 점검하였다. 이 단계에서는 실제 커스터마이징 언어인 Visual Basic을 가지고 실제로 사용할 화면을 구성해 나갔다.

위 Validating Model의 두 단계 완료 이후, 시나리오에 근거하여 화면만으로 구성된 GUI Prototype을 만들어 이를 가지고 다시 계속적인 시스템 구현 이미지에 대해 검토하였으며, 이는 자칫 이해하기 힘든 실제 시스템의 명확한 이미지를 제시해 줄 수 있기

때문에 매우 좋은 효과를 볼 수 있었다. 이를 가지고 고객과 개발 벤더간의 시스템 설계서에 대한 공유가 가능하였고, 이 후 정의된 화면들과 제약들은 시스템의 DB Design을 자연스럽게 유추할 수 있어 굳이 기본적인 DB 설계를 다시 할 필요가 없게 되었다.

㉔ 그 밖의 설계 산출물 정의 및 도출 방법

이상과 같은 시나리오 역할 게임을 통해서 고객과 개발 벤더간의 이미지 공유 및 설계 Spec.의 정의가 가능하였으나, 이 외에도 시스템의 설계를 위해서는 유추해야 할, 혹은 추가적으로 검토해야 할 사항들이 있다. 이들은 주로 해당 패키지에 따라 파악되어야 할 구체적 항목은 차이가 있을 수 있겠으나 대체로 유사한 범위 내에서 요약될 수 있으며, 이들을 추가하여 제시한 시스템 설계서의 목차는 다음과 같다.

- 목차 -

1. 프로젝트 정의
2. 프로젝트 추진 조직
3. 프로젝트 추진 범위
4. 프로젝트 추진 일정
5. 시스템 설계 단계 산출물
 - 5.1 관리항목의 정의
 - 5.2 시스템 모듈별 기능 정의
 - 5.3 시스템 구성 정의
 - 5.4 IMAN Object Design
(패키지 시스템 데이터 모델)
 - 5.5 화면 설계 (GUI Object Class Card)
 - 5.6 사용자 동작 시나리오 정의 (IDEF3 모델)
 - 5.7 관리번호 부여기준의 정의
 - 5.8 메시지 표준 정의
 - 5.9 Application Encapsulation 정의
(타 시스템과의 Interface/Integration 정의)
 - 5.10 사용자별 권한 범위 정의
(보안관리 측면의 정의)
 - 5.11 사용자별 운영 환경 정의
 - 5.12 AutoCAD Application 설계
(추가 시스템에 대한 별도 정의)
6. 개발 및 구현계획
 - 6.1 상세 구현 계획
 - 6.2 테스트 수행 계획
 - 6.3 기 보유 데이터 이관 계획
 - 6.4 교육 지원 계획

위 내용 중 패키지 시스템의 데이터 모델은 GUI Object Class Card를 기준으로 유추하여 정의하고 이에 대한 형식이나 크기 등은 이미 정의/분석된 Data-Process Transaction Diagram을 참조로 하여 쉽게 정의할 수 있었다. 또한, 사전 검증된 시나리오에 대해서 명확히 결정지어진 시스템의 범위에 따라 최초 구성한 구현 계획을 다시 개발자 능력별로 조정하여 재구성하고, 변경사항에 대해 고객과 개발 벤더간의 합의를 하여 프로젝트 납기 및 일정의 변경 가능성을 최소화 시킬 수 있었다. 단, 시스템을 사용할 모든 부서의 부서원들이 실제 이 역할 게임을 한 것이 아니기 때문에, 또한 현업의 프로세스가 개발 도중 변경되었기 때문에 일부 정정작업이 필요하였으나, 이미 시스템에 대해서 고객이 충분히 알고 있기 때문에 변경관리가 훨씬 용이해졌다. 시스템 동작 시나리오까지 정의라고 이에 대해 시스템 Fine Tuning 까지 수행하려고 하였지만 이에 대해서는 시간적 제약으로 개발자 상호간의 목시적인 작업으로 수행함을 대처하였다.

이상과 같은 작업은 반드시 사용될 패키지에 대해서 명확한 이해를 하고 있는 개발 벤더의 리더가 그 제약점과 효과적인 측면을 잘 고려하여 제시할 수 있어야 하며, 자칫 잘못된 방향의 설정으로 위 내용을 수행한다면 오히려 고객과 개발 벤더간의 큰 오해를 가져올 수도 있다는 점을 명심해야 할 것이다.

4. 결론 및 추후 연구방향

이상과 같은 방법으로 본 연구 대상이 된 PDM 응용 프로젝트는 성공적으로 추진될 수 있었다. 물론 기간상의 문제로 인해 매우 상세한 단계까지의 시스템 설계에 대한 검증이 있을 수 없었으나, 위 방법을 통해 고객과 개발 벤더간의 시스템 이미지 공유는 물론 시스템 설계자와 개발자간의 설계 공유를 위한 도구로서 매우 효과적인 역할을 훌륭히 수행할 수 있었다고 자신한다.

이 방법은 앞서 언급한 바와 같이 설계단계에서 미리 현업에서 느끼는 업무 프로세스에 맞는 실행을 해 봄으로써 발생할 수 있는 문제점에 대한 검증이 가능하며, 수 차례 되풀이하여 진행됨으로써 더욱 더 명확한 상호간의 최적 해를 사람이 가장 알기 쉬운 GUI로써 구성할 수 있다는데 그 장점이 있다. 또

한, 패키지 형태의 이미 갖춰진 데이터 구조에 일부 클래스를 더하거나 활용하는 부분에 있어서는 GUI Object class Card에서 도출된 각 Attribute와 Action에 따른 연계를 통해 쉽게 개발자가 파악/정의 할 수 있어 DB Design 까지도 자동적으로 유추해 낼 수 있다는 점에서 더 실용적이다.

본 방법의 활용을 통해 많은 시스템 개발시 신속한 설계와 합의를 통한 안정된 시스템 구현에 기여할 수 있으리라 생각되며, 아울러 본 방법은 개략적인 업무범위가 확정된 상태에서 실제 시스템 설계에 유용한 것으로, 개발측에서 패키지 시스템에 대한 명확한 파악이 선행되어 있을 경우를 기준으로 하였으나, 이 외의 기존 개발형태에 있어서도 고객과의 합의 도출을 위한 방법으로는 많은 효과가 있으리라 확신한다.

참고문헌

1. 최영근, 허계범, 객체지향 소프트웨어 공학, 한국 실리론, 1995.
 2. David A. Taylor, PhD., Business Engineering with Object Technology, John Wiley & Sons, Inc., 1995.
 3. Hans-Jorg Bullinger, Dr-Ing, Dr-Ing habil, Joachim Warschat, Dipl-Ing, Dr-Ing, Concurrent Simultaneous Engineering Systems, Springer, 1996.
 4. LG-EDS 시스템 기술 연구부문/기술확산팀, C/S 10,000을 이용한 프로젝트 관리방법, 1997.
 5. Richard J. Mayer, PhD., IDEF0 Function Modeling, Knowledge Based Systems, Inc., 1994.
 6. Richard J. Mayer, PhD., Michael K. Painter, Paula S. deWitte, PhD., IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications, Knowledge Based Systems, Inc., 1994.
 7. Richard J. Mayer, PhD., Thomas P. Cullinane, PhD, Paula S. deWitte, PhD., William B. Knappenberger, Benjamin Perakath, M. Sue Wells, IDEF3 Process Modeling, Knowledge Based Systems, Inc., 1994.
1. 최영근, 허계범, 객체지향 소프트웨어 공학, 한국 실