

Payment Settlement Framework for Exporting Real-Time Online Financial Solution

Huynki Bae* · Yunji Ahn** · Kwangho Park***†

*Department of Business Consulting, Graduate School of Hanyang University

**DIREA

***Department of Business Administration, Hanyang University

실시간 온라인 금융솔루션 수출을 위한 지급결제프레임워크

배현기* · 안윤지** · 박광호***†

*한양대학교 일반대학원 경영컨설팅학과

**주디리아

***한양대학교 경상대학

Korean small and medium sized software companies have tried to export their solutions or services to overseas markets. In 2016, exports of the software industry increased by 6.0% from the previous year, and the value added of the industry was 2.2 times higher than that of the manufacturing industry. From a long-term perspective, it is important to secure a global competitive advantage in order to sustain the export high value-added of the software industry. The obstacles to entry into the overseas market of small to medium enterprises are as follows: first, difficulty in product development and localization of marketing; second, lack of investment for overseas expansion; and finally, competitiveness of software technology. In particular, To overcome such obstacles, Korean small and medium sized software companies should increase the technical perfection and secure software export competitiveness. The paper presents a payment settlement framework enabling adaptive reuse and semiautomatic development of global payment settlement services. The quantitative and qualitative evaluation results are presented with domestic and overseas case studies as follows: Firstly, semi-automatic development is realized successfully by applying the framework. Secondly, it is possible to maintain consistent quality of software and to deliver maintenance services without relying on the internal human resources. Thirdly, it is possible to reduce the project duration of the same development cope to less than 50% by applying the framework. Finally, because it is based on BPMN 2.0, which is a high level design diagram, it is expected that it will be easy to implement through components connection and reduce difficulties in technology transfer and localization. Also, at the time of runtime operation, it will be effective to understand the design idea easily and to carry out additional developments without human resource who participated in the initial project.

Keywords : Payment Settlement Framework, Automated Clearing House, On Line Banking System

Received 20 November 2017; Finally Revised 13 December 2017;

Accepted 14 December 2017

† Corresponding Author : oobepark@hanyang.ac.kr

1. 서론

가치창조의 실현이 소프트웨어를 중심으로 혁신되고 있다. 한국은행[19]의 조사에 따르면 2014년 소프트웨어 산업의 부가가치율은 제조업(23.6%)의 2.2배인 49.0%로 고부가가치를 창출하는 미래 산업이다. 특히 2016년 소프트웨어 산업의 수출액은 전년대비 6.0% 증가한 63.9억 달러로 전 산업의 수출액이 감소한 것에 비해 성장했다[4]. 그렇지만 글로벌 경기침체로 인해 2016년 IT서비스 수출은 전년 대비 9.8% 감소한 29.2억 달러를 기록했다[3]. 내수 시스템통합(SI)사업의 성장 정체로 해외진출이 적극적으로 진행되고 있으나 대형 IT서비스 기업들을 중심으로 시장이 확대되고 있다. 반면 중소 소프트웨어 기업의 성과는 저조하다. 이러한 원인은 한국 소프트웨어 산업의 구조적 문제에서 찾을 수 있다. IT서비스는 산출물의 특성상 인력 중심의 프로젝트로 수행되기 때문에 내부 개발보다 아웃소싱을 통해서 진행되고, 관행적으로 소프트웨어를 비용절감의 대상으로 인식한다[17]. 이로 인해 중소 소프트웨어 기업들은 전문 인력 양성, 소프트웨어 자산화 그리고 품질 개선보다 비용 절감에 우선순위를 두고 있어 기술력과 전문성을 축적하지 못하고 있다.

어떻게 하면 한국 중소 소프트웨어 기업들이 기술완성도를 높여서 소프트웨어 수출 경쟁력을 확보할 수 있을까? 소프트웨어 기술경쟁력을 높이기 위해 소프트웨어 재사용이 오랫동안 연구되어 왔다. 소프트웨어 재사용을 통한 개발의 효율성, 품질의 일관성, 노하우의 자산화, 유지보수 편의성 등이 시도되어왔고, 재사용의 일환으로 소스코드의 컴포넌트화 및 이를 기반으로 한 프레임워크 방법이 연구 및 활용되고 있다[2, 6, 7, 8, 11, 13, 16].

하지만 기존의 연구는 시스템 일반적인 부분과 특정 비즈니스 공통적인 내용을 컴포넌트화 하거나, 특정업무 전용 패키지화 부분에서 많이 진행되었는데, 반자동화 개발과 적응형 재사용을 염두에 둔 프레임워크 부분에 대해서는 많은 연구가 진행되지 않았다. 본 연구의 목적은 첫째, 소프트웨어 재사용관점에서 적응형 재사용과 반자동 개발이 가능한 프레임워크 설계의 요구사항을 도출하고, 둘째, 글로벌 지급결제청산 업무 개발을 위한 BPMN (Business Process Management Notation) 2.0 기반의 프레임워크를 설계하여, 셋째, 프로토타입 개발 및 사례연구를 통해 평가, 개선하고자 한다. 본 연구 결과를 통해 해외 시장에 진출하는 중소 소프트웨어 기업이 특히, 금융 IT서비스 분야와 같이 인력에 대한 의존도가 높은 시스템 구축 사업을 추진할 때, 프로젝트에 대한 경험과 노하우가 소프트웨어 자산에 축적된 패키지 소프트웨어를 기반으로 기술경쟁력을 향상시킬 수 있을 것이라 기대한다.

2. 이론적 배경

2.1 중소 소프트웨어 기업의 해외 시장 진출 장애요인

선행연구에서 제시된 중소 소프트웨어 기업의 해외 시장 진출 장애요인은 다음과 같다.

첫째, 현지화 어려움이다. 현지 문화나 환경에 맞춘 마케팅이나 상품개발을 진행하기 위해서 거점 설치 및 현지 파트너와의 협력을 통해 현지화 된 서비스 제공이 필요하다. 국가 간 심리적 거리의 차이가 크지 않더라도 현지화를 위해 직접 진출 방식을 선택해야 하는 경우가 많기 때문이다. 대표적인 대상은 기업용 솔루션, 전자정부 솔루션 등 도메인별 특성을 반영한 소프트웨어 제품들로, 이를 해외에 판매하기 위해서 현지화가 필요할 뿐만 아니라 판매 후 서비스도 제공되어야 한다. 또한 국가별로 솔루션 소프트웨어에 적용되는 산업의 특성이 상이할 수 있기 때문에 솔루션 현지화가 필요할 수 있다[16].

둘째, 해외 진출을 위해서는 과감한 투자가 필요하다. 해외거점 확보 등 다양한 투자가 필요하지만 중소 소프트웨어 기업 규모로 대응이 어려운 상황이다. 자금력으로 해외진출을 위한 자체적 판매망의 구축이나 서비스망의 완비가 불가능하기 때문에 체계적인 해외 조직 구축 보다는 기회가 생기면 진출을 타진하는 소극적 자원 배치에 머물고 있다[16].

셋째, 국내기업의 취약한 소프트웨어 기술경쟁력이다. 국내 IT산업 경쟁력 하락의 원인으로 소프트웨어의 경쟁력 부재가 가장 많이 제시되고 있다. 한국 소프트웨어 산업 경쟁력은 OECD 19개국 중 14위에 불과하다고 발표된 바 있으며, 소프트웨어 개발역량은 후진국 수준으로 평가되고 있다. CMMI(Capital Maturity Model Integration) 소프트웨어 품질지수는 인도 4.30, 미국 3.06, 중국 2.97, 한국 2.94으로 조사되었다. CMMI는 소프트웨어 품질보증 기준으로 널리 사용되고 있으며, 1~5단계로 소프트웨어의 품질을 평가한다[14]. 소프트웨어 분야 중에서 특히 패키지 소프트웨어는 가격 민감성보다 성능 및 품질 민감성이 높으며, 브랜드와 품질경쟁력이 있는 글로벌 우수 제품으로의 치우침이 높은 분야로 독창성과 우수한 기술력 없이는 해외진출이 매우 어렵다[9].

따라서 본 연구에서는 국내 소프트웨어 중소기업의 해외 진출 장애요인 중 기술경쟁력 약화에 초점을 맞춰서 원인을 분석하고자 한다.

2.2 중소 소프트웨어 기업의 기술경쟁력 약화 요인

중소 소프트웨어 기업의 기술경쟁력 약화 요인에 대한 선행연구는 다음과 같다.

첫째, 하드웨어 중심의 산업 구조다. 한국의 대기업들은 산업화 기간 동안 하드웨어에 치중해 있었고 상대적으로 소프트웨어는 관심이 적었다. 소프트웨어 산업을 국가적으로 육성하려고 정부에서 지원했지만 기반이 약하여 성장 속도가 늦었다. 소프트웨어 경쟁력은 하드웨어를 포함한 IT산업 전반의 경쟁력 약화로 이어졌다[17].

둘째, 소프트웨어 활용도 부족이다. 각 산업의 생산물에서 소프트웨어 활용률이 선진국 대비 현저하게 낮은 편인데, 만약 소프트웨어 활용률을 선진국 수준으로 높인다면 GDP 증가에 많은 기여를 할 수 있다. 활용률이 낮은 원인으로 소프트웨어 기술 수준이 낮다는 것과 전문 기업의 부재가 있다[14].

셋째, 인력 중심의 프로젝트 수행과 소프트웨어 자산화 취약 문제가 있다. 국내 기업들이 보유한 많은 경험과 지식을 자산화 하지 못하고 인력에 의존하는 경향을 보이고 있다. 대부분 공공기관이나 발주 주체인 대기업들의 소프트웨어 구입은 제품 가치를 인정하여 대가를 지불하기보다 단순 노동력을 대가로 거래되는 용역형태로 진행되어 왔다. 이렇게 거래된 원도급자도 많은 부분을 재하도급으로 업무 이관하기 때문에 소프트웨어 개발 기술과 노하우를 소프트웨어 기업이 보유한 패키지나 솔루션에 축적하기가 어려운 상황이다[14].

본 연구에서는 국내 소프트웨어 중소기업이 보유한 경험과 지식에 대한 자산화 이슈를 소프트웨어 재사용 관점에서 프레임워크를 확보하는 방안으로 해결안을 도출하고자 한다.

2.3 소프트웨어 재사용 관점의 프레임워크에 대한 연구

이상적 재사용 기술은 컴포넌트를 사용하여 새로운 시스템을 구축할 때 컴포넌트 간의 연결을 용이하게 하는데 있다. 소프트웨어 개발자는 컴포넌트의 내부적 구현 방법을 파악할 필요 없이 외부적 사용 방법만 익혀서 사용할 수 있다. 결과적으로 컴포넌트를 활용하여 구축된 시스템은 보다 효율적이고 유지보수가 용이하며 안정적이라고 볼 수 있다[1]. 박광호[15]의 연구에서 제시된 바와 같이 재사용 대상 선정 방법에 따라 재사용 시도의 성공 여부가 달려있다고 가정하고 <Figure 1>에서처럼 기본기능, 관리지원기능, 업무처리기능으로 분류해 볼 때, 전체의 79%에 해당하는 기본기능과 관리지원기능을 재사용 대상으로 분류한다.

또한 프레임워크의 유형을 수평적 프레임워크와 수직적 프레임워크로 구분하였다. 수평적 프레임워크는 다양한 애플리케이션에 적용할 수 있는 전문지식을 추상한 것으로 사용자 인터페이스(User Interface : UI), 데이터베이스(Database)

			Database
			Business Logic
			User Interface
Basic Function(56%)	Management Support Function(23%)	Business Process Function(21%)	

<Figure 1> Ratio of Program Code

4 Layer(5% Under)
3 Layer
2 Layer
1 Layer(50~80%)

<Figure 2> Frame Structure for Adaptive Reuse

스(Database) 등의 애플리케이션에 독립적인 기능들을 수평적으로 분할한 구조이다. 수직적 프레임워크는 특정 문제 분야의 전문지식을 캡슐화한 것으로 제조, 금융, 서비스, 의료분야 등 수직적 분할 구조이다. 수직적 프레임워크를 구성하는 객체를 ‘비즈니스 객체’라 부르는데 비즈니스 도메인에서 활동하고 있는 객체들을 표현한 것으로 비즈니스 이름, 정의, 속성, 행동, 관리, 제약조건 등을 포함한다.

Bassett[1]이 제시한 적응형 재사용은 일반화된 구성요소(Generalized Component: 프로그램 서브루틴 객체 등)에 맞게 조정하는 프로세스를 말한다. 이는 환경의 변화를 인식하고 운용 상태를 평가하여 스스로 동작한 행위를 환경에 맞춰 대응하도록 설계하는 것을 의미하기도 한다[11]. 적응형 재사용을 위한 프레임(Frames) 4계층구조는 <Figure 2>와 같다.

1단계는 데이터베이스, 윈도우, 리포트, 네트워크 프로토콜, OS API 등을 의미하며, 2단계는 UI 표준, 오류 감지 및 복구, 보안, 로그인 절차 등을 처리하는 비즈니스에 독립적인 계층이다. 은행의 경우 3단계는 계좌 및 금리관리와 같은 비즈니스 규칙 프레임에 의미한다. 4단계는 응용프로그래머가 작성하고 수정할 수 있는 상황에 맞는 계층으로 구분하였다[1].

Frakes and Kang[2]는 재사용을 특정한 영역(Domain)에서 특화된 형태로 반복적 구축하는 것이며, 대부분의 소프트웨어 시스템은 새로운 것이 아니라 기본 관점을 갖고 있다. 이는 박광호[16]의 수직적 프레임워크 이론과 같은 맥락으로 본 연구에서 다루고자 하는 지급결제 영역 특화 프레임워크 개발의 당위성이 제공된다고 볼 수 있다.

소프트웨어 재사용과 이를 기반으로 한 프레임워크에 대해 많은 연구가 진행되었다. 특히, 재사용 효과가 높지 않고 실패하는 경우가 많이 발생된다는 연구결과가 제시되었는데, Frakes and Kang[2]은 그에 대한 원인을 다음과 같이 다섯 가지로 설명한다.

첫째, 지속성 문제이다. 현재 다양한 산업에 많은 재사용 프로그램들이 존재한다. 그러나 장기간 지속가능한 재사용 프로그램을 찾는 것은 어렵다. 이 부분은 재사용 프로그램과 도메인 엔지니어 그리고 기업의 전략 연계 방법이 중요한 문제이다. 재사용 대상과 이를 통해 발생하는 수익적 측면의 효과를 고려해야 한다.

둘째, 기술 전수 문제이다. 재사용 개발자가 도메인 엔지니어에게 시스템에 대한 부분과 비즈니스에 대한 부분을 잘 전달하는 것이 중요하다.

셋째, 기업의 전략은 지속성 부분에서 언급한 것과 같이 전략이 공유되어야 수익성이 높고 유지 가능한 재사용 프로그램이 생산될 수 있다.

넷째, 조직적 접근 방식의 이슈이다. 중앙집중식 접근은 개발, 판매, 유지보수, 교육에 대한 조직적 집합을 구성하여 시작하는 방법이다. 이러한 방식은 개발의 비전이나 목적을 공유하고 전문 인력들을 체계적으로 활용하고 개발되는 자산이 체계적으로 관리 되는 장점이 있으나 초기 투자비용이 많이 들고 개발자들을 실제 프로젝트에서 차출해야 하는 단점이 있다. 반면 분산적 접근은 실제 프로젝트와 연계해서 재사용 프로그램을 구축한다. 즉 각각의 프로젝트에서 재사용과 관련된 기회를 발견하여 개발해나가는 방법이다. 이러한 방법의 단점은 첫째, 만약 정확한 비전이 없다면, 각각에서 개발한 자산들을 조합하는데 어려움이 있을 수 있다. 둘째, 공유되는 비전이 존재해도 프로젝트에서 개발된 컴포넌트가 다른 프로젝트의 요구를 정확히 맞추기 어렵다. 싱크를 맞추고 트랜잭션을 관리하고 전체를 구성하기 위한 비용이 발생할 것이다.

다섯째, 프로세스 초점에서 볼 수 있다. 프로세스 초점은 적극적인 투자(Proactive investment)와 반응형 투자(Reactive investment)로 구분된다. 적극적인 투자는 목표도메인 또는 제품 라인에 대한 분석과 도메인의 아키텍처에 대한 정의를 완료하고 개발을 시작하는 방법으로 초기에 투자비용이 많이 소요될 수 있다. 반응형 투자는 개발 중에 재사용이 가능한 부분이나 문제를 해결할 수 있는 부분을 자산화 하는 방법으로 초기 투자비가 적게 드는 장점이 있으나 향후 제품에 대한 지속적인 수정작업이 필요하다는 단점이 있다[2].

프레임워크 실패 원인의 다른 연구로 Jonson[6]의 연구가 있다.

첫째, 프레임워크는 지속적으로 재사용되어야 한다. 프

레이워크를 적용하여 기술력을 향상시키고 효율성을 높일 수 있어도 장시간 재사용하는 것이 중요하다. 프레임워크는 효과는 강력한 반면에 사용이 복잡하기 때문에 프레임워크를 정확히 이해하는 것이 매우 어렵다. 그렇기 때문에 프레임워크를 지원할 수 있는 매뉴얼이 필요하며, 사용자에게 교육기간이 제공되어야 한다. 또한 프레임워크 개발을 위해서 타 시스템 대비 더 많은 시간과 자원이 투입되어야 한다.

둘째, 개발 언어에 대해서 언급하였다. 과거 프로젝트에 여러 언어가 사용되었으나 최근 개발 환경은 대부분 자바의 J2EE(Java 2 Enterprise Edition) 기반으로 개발되고 있다.

2.4 기술경쟁력 강화를 위한 프레임워크의 고려사항

중소 소프트웨어 기업의 해외진출 장애요인과 기술경쟁력 약화 원인 그리고 소프트웨어 재사용을 위한 프레임워크 관련 선행연구를 분석한 결과 본 연구에서 제시할 지급결제프레임워크(Payment Settlement Framework : PSF)에 적용할 기술적 고려사항은 다음과 같다.

첫째, Jonson[6], Frakes and Kang[2] 그리고 박광호[13]의 연구에서 제시하는 수직적 구조, 특정한 영역에 특화된 프레임워크로 지급결제 영역을 위한 프레임워크를 개발한다. 이를 통해 프레임워크의 복잡도를 최소화한다.

둘째, 적응형 재사용을 위한 구조로, 런타임(Run-Time) 아키텍처 개념을 적용하여 런타임 중에 변경이 생길 수 있는 요소에 대해 추가, 수정, 삭제가 가능하도록 한다. 이는 컴포넌트만이 아닌 파라미터 설정을 통해서 수행하는 구조도 포함한다. 지급결제 시스템은 구성상 외부 시스템과 연계가 많고, 연계 방식이나 이에 따른 메시지 형식도 다양하다. 또한 금융 서비스라는 비즈니스 업무의 특성상 새로운 상품 또는 서비스가 수시로 생겨나고 없어지기 때문에 이에 따른 런타임 개념을 아키텍처에 도입하는 것은 중요하다[1, 20].

셋째, 재사용의 지속성과 기술이전의 용이성을 고려한다. 이를 위해 우선 컴포넌트 저장소를 전사적으로 구성하고, ISO 20022, ISO 8583 등 지급결제 분야를 위한 컴포넌트들의 글로벌 표준을 준수한다. 또한 BPMN 2.0 기준을 적용한 시각적 설계로 개발자들 간의 기술 전수 뿐만 아니라 글로벌 사용자들에게 운용을 위한 기술이전을 쉽게 한다.

넷째, 실시간 온라인금융 시스템 환경과 해외 수출을 위한 프레임워크 요구 사항으로 반자동 개발 개념을 적용한다. 중소 소프트웨어 기업의 수출 장애요인에서 기술경쟁력 취약의 원인으로 인력 투입 비중이 높다는 것과

이로 인한 가격 경쟁력이 낮아지는 것, 관련 산업의 경험과 노하우가 소프트웨어 자산화 되지 못하고 개발자들에게만 쌓이는 문제가 있었다.

본 연구에서는 지급결제분야에서 필요한 컴포넌트 들을 중앙집중식과 분산형 개발을 병행하여 고객 현장에서 프로그래밍(On Site Program)이 없어도 되는 개발이 가능한 프로세스를 구현하고자 한다[2]. 이를 위해 준비된 프레임워크와 컴포넌트 저장소의 일반요소(Generalized Component)와 특수요소(Specialized Component)를 활용하여 고객 현장에서는 시각화 설계 도구를 이용하여 설계하는 것만으로 실행모듈을 생성할 수 있는 반자동 개발 프레임워크를 구현한다.

반자동 개발을 위해서는 첫째, 소프트웨어 재사용을 위한 중앙집중식 조직이 필요하고 고객 현장에서 신규 도출되어 기존 컴포넌트 저장소에 존재하지 않는 컴포넌트들을 중앙 집중 조직에서 개발하고 테스트하여 프로젝트 현장에 공급한다.

둘째, BPMN 2.0 기반으로 설계한 비즈니스 프로세스가 설계 문서에 그치지 않고 실행 모듈 생성과 연결되어 지급결제 업무를 위한 프로그램이 생성되도록 BPM 엔진을 개발하고 이것을 메시지 처리 엔진(Message Processing Engine)과 연동하여 구현한다.

3. 글로벌 지급결제 프레임워크 제시

3.1 BPMN 2.0 적용을 위한 설계

PSF의 중요한 기본 기능은 실시간 온라인 트랜잭션 처리이다. BPMN 기반으로 설계된 프로세스를 통해 어플리케이션(Application)을 개발하기 위해서 3가지 요인을 프레임워크 설계에 반영하고자 한다.

첫째, ESB(Enterprise Service Bus) 엔진과 BPM 엔진을 연동한다. J2EE 기반의 ESB 엔진을 사용하고, ESB 엔진과 연동하여 모든 비즈니스 트랜잭션을 프로세스 관점에서 처리해 주기 위해 BPM 엔진을 사용하여 구성한다. 또한 엔진 간 인터페이스를 위해 전문 규약을 설계하고 인터페이스 모듈을 개발한다.

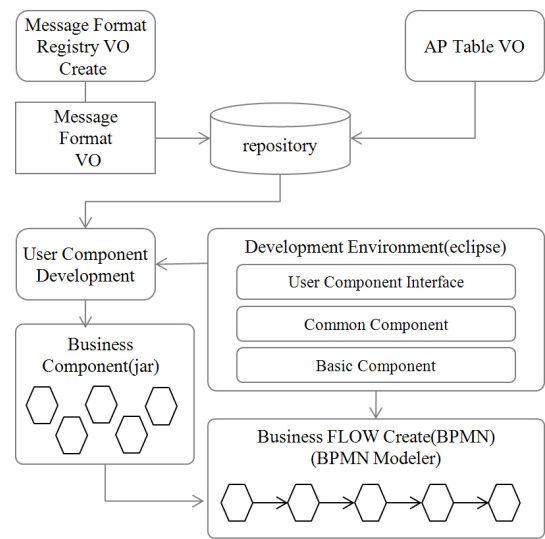
둘째, ESB 엔진에서 BPM 엔진으로 전달된 메시지를 처리하기 위해서 BPM 서비스를 호출하는 BPM 매니저(Manager)를 개발한다. ESB 엔진에서 BPM 엔진으로 전달된 메시지를 처리하는 절차는 다음과 같다. 우선 인터페이스부의 채널 어댑터를 통해 실시간 전문 혹은 배치 파일을 수신하면 전문으로부터 업무 정보를 확보하고 BPM 매니저를 호출한다. 다음으로 호출된 BPM 매니저는 업무정보에서 취득한 정보를 기반으로 BPM 기반으

로 설계된 프로세스를 탐색한 후 실행시킨다.

셋째, 설계 단계에서 글로벌 표준인 BPMN 2.0 노테이션(Notation)을 기반으로 설계하고, 설계된 프로세스로 실행 모듈이 생성되도록 한다. 이를 통해 기술이전, 지속적인 재사용[19], 그리고 반자동화 개발을 실현한다.

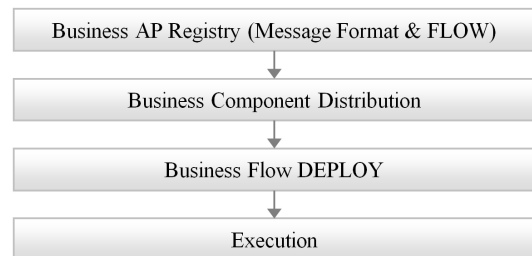
BPMN 2.0 기반의 개발 아키텍처는 다음과 같다.

우선 사용자 정의 비즈니스 태스크(Business Task)를 생성하고 BPMN 모델러(Modeler)의 태스크 및 공통 컴포넌트를 이용하여 플로우(Flow)를 생성하며 <Figure 3> 같이 표현될 수 있다.



<Figure 3> Step of Business Process Creation

컴포넌트 배포와 프로세스 실행절차는 <Figure 4>와 같다. 생성된 비즈니스 클래스와 모델링 된 BPMN XML (eXtensible Markup Language)을 서버에 배포하고 BPM 엔진에서 공통 컴포넌트를 통해 클래스 헤더(Class Header)에 로드된 BPM 프로세스를 실행한다.



<Figure 4> Component Distribute & Process Execute

BPMN 2.0 기반으로 설계된 프로세스를 실행시키기 위한 Deploy Method는 <Table 1>, Start Method는 <Table 2>와 같다.

<Table 1> BPM Flow Deploy Method

Name	Deploy method(BPMDeployManager)
Location	Group/domains/mule-domain-config.xml
Script	<spring:bean id="BPM_DEPLOY" class="kr.co.direa.server.cruzlink.bpms.com. BPMDeployManager" init- method = "onCall"/>

<Table 2> BPM Process Start Method

Name	Start method(BPMEngnManagement)
Location	Group/aos/default/mule-config.xml
Script	<custom-filter name="WorkflowManager" class="kr.co.direa.server.cruzlink.bpms.com.BPMManager"/> <filter ref="WorkflowManager"/>

BPM 프로세스를 서버 기동 시에 클래스 로더에 적재하기 위하여 서버 기동 환경설정 파일에 Spring Bean으로 서버 시작 시 실행되어 적재하도록 설정하며, 프로세스 별로 호출하기 위한 세션 값을 전역변수에 저장한다. 적재되어 있는 BPM 프로세스를 거래전문 요청 수신 후 해당 프로세스를 찾아 기동하는 컴포넌트를 서버 환경설정 파일에 설정한다. 전문 수신 시 해당 플로우를 수행하는 동시에 Workflow Manager를 수행하여 BPM 프로세스를 진행하는 API를 호출한다.

3.2 적응형 재사용을 위한 설계

적응형 재사용이란 일반화된 구성요소(Generalized Component)를 다양한 환경에 맞게 조정하는 프로세스를 의미한다.

런타임 컴퓨팅 개념을 적용하여 운용 중에 수시로 변경되는 비즈니스 요건에 따라 컴포넌트 및 관련 플로우가 자동으로 가변 적용 되거나[11, 20] 환경 변수를 파라미터 설정 변경하는 것으로 요건 변경에 대응 하도록 한다. 신규 업무 요건이 발생하는 상황에서도 기존 서비스를 중단하지 않은 상태에서 새로운 BPM 프로세스를 실행할 수 있도록 지원한다.

실시간 온라인 금융공동망 환경에는 개인, 기업, 금융기관, 결제기관 간에 전문 형식, 통신 프로토콜, 비즈니스 서비스 등 빈번히 발생하는 변화 요소 들이 많다. 이 요소 들을 시스템 운용 중에 무정지(Non-Stop) 상태 동적 재구성(Dynamic Reconfiguration)할 수 있도록 설계한다.

지급결제시스템에 새로운 비즈니스 요건이 발생하여 신규업무를 적용하는 경우에 인터페이스 환경이 변경하게 되는 경우를 예로 동적 재구성 절차는 <Table 3>과 같다.

<Table 3> Interface Dynamic Reconfiguration Process

Step	Detail
1) Channel Information Enrollment	- Channel Information(Group, Encryption/Decryption, Template, Channel Code, Channel Name, Institution Name, Backend Channel, Adapter Information) Enrollment - Classification Standard Enrollment(offset, Length Value Enrollment) - Standard Information Registration
2) Message Information Management	- Register Message Information in Excel - Mapping Request Message/Response Message - Saving Mapping Rules as Stored in DB
3) Real-Time Business Management	- INBOUND Transactions-Registering CORE SVC Code - Registering Fixed Message Standard - Registering Encoding Standard
4) Maneuver Management	- Maneuvering by Groups, Channels, Batch Shells - Management Maneuvering by Groups, Channels in Service Detail View-Detail List

런타임 중 해당 컴포넌트에 반영되는 파라미터(Parameter) 값을 동적으로 적용하여 서비스에 반영하는 것으로 적용 파라미터 변수는 첫째, 데이터 타입, 클래스 명, 인코딩 타입, 암호화키, 메시지 라우팅, 로그 설정 파일 설정 등의 기본 설정, 둘째, 망상태 관리, 거래내역 관리, 예외처리 관리 등의 공통 컴포넌트에 적용되는 파라미터가 있다.

PSF를 사용하여 용도별로 구축하게 되는 지급결제시스템에서 업무별 특성이 있는 컴포넌트에 반영되는 파라미터는 이용기관관리 전문 변환 관리 등이 있다.

3.3 컴포넌트 구성

PSF의 전체 컴포넌트 구성은 <Figure 5>와 같다.

첫째, 기본 컴포넌트(Basic Component)는 시스템과 관련된 처리를 수행하기 위한 것으로 파일 컴포넌트, 전문 컴포넌트, 데이터베이스 컴포넌트, 네트워크 컴포넌트, 트랜잭션 컴포넌트, 총 5개 컴포넌트 그룹으로 구성된다.

둘째, 어플리케이션 공통 컴포넌트(Application Common Component)는 어플리케이션에서 공통으로 사용하는 컴포넌트들로 사용자관리, 수수료(Fee)관리, 과금(Billing)관리, 인증(Validation)관리, 망상태관리, 결제관리, 거래내역관리, 예외 처리 관리 총 8개 컴포넌트 그룹으로 구성된다.

셋째, 서비스 사용 대상별로 비즈니스 어플리케이션에서 변경되는 컴포넌트(Application Variable Component) 들로 전문 변환 컴포넌트와 파일 변환 컴포넌트, 국가별 계좌관리 컴포넌트 총 3개 컴포넌트 그룹으로 구성된다.

넷째, 결제청산시스템(Automated Clearing House Framework : ACHF), 펌벙킹시스템, 전자금융공동망시스템 등

App. Specific Component	ACH-Central Comp. •Net Debit Cap •Mandate •Dispute •Reconcile •Clearing	ACH-Member Comp. •Local Debit/Credit total mgt. •Transaction Input •Details Management	Firm-Banking Comp. •Institution Management •Deposit/Withdrawal Details •Decomposition/Gathering by Institution •scheduling	Joint Network Comp. •Fund Return •Account Link •incomplete treatment	
App. Specific Component	Message transfer Comp. •ISO 20022 Transfer •ISO 8583 Transfer •XML/JSON Transfer •Fixed Transfer	File Transfer Comp. •ISO 20022 Transfer •XML Transfer •Fixed Transfer	Account by Country Comp. •Account •Phone No Account		
App. Common Component	Network State Mgt. •Inquiring Network Status	Approval Mgt. •Approval List •Approval Approve •Approval at Current State	Exception Handling Mgt. •Resend •Cancel •Reprocessing •Alarm	Transaction History Mgt. •Total/Statistics	Classification Mgt. •Sorting •Decomposition •Gathering
	User Mgt. •individual/Enterprise •Validation •Insert/Modify/Delete	Fee Mgt. •Strategy •Real Time Commission Calculation •Statistics/Total	Billing Mgt. •Aggregate Management •Asset Management	Validation Comp. •File Format •Field Format •Valid Value Validation	Encryption/Decryption Comp. •Signature •Encryption /Decryption by Field •Total Encryption /Decryption
Basic Component	File Comp. •File Load •File Create •Move/Copy •Rename	Message Comp. •Parsing •Assemble •Transform	DB Comp •Connect/Release •SQL perform	Network Comp. •File Send/Receive •Message Send/Receive	Transaction Comp. •TR Start •Full commit •Partial commit

<Figure 5> PSF Component Architecture

비즈니스 서비스마다 독특하게 사용하는 비즈니스 어플리케이션 컴포넌트들(Application Specific Component)로 ACH-Central 컴포넌트, ACH-Member 컴포넌트, 펌뱅크 컴포넌트, 공동망 컴포넌트, 총 4개 컴포넌트 그룹으로 구성된다. 이곳에 새로운 지급결제시스템 유형이 생겨날 때마다 신규의 컴포넌트 그룹을 생성하여 저장소에 보관하게 된다.

3.4 프레임 구조에 따른 PSF 아키텍처

어플리케이션에 독립적인 기능들을 수평적으로 분할한 프레임워크 구조와[15] 계층적 기능 구분에 따른 프레임 구조[1]의 연구를 반영하여 설계된 PSF의 구조는 <Figure 6>과 같다.

첫째, 연계 인터페이스 모듈은 엔진, 파라미터 설정부, 컴포넌트의 세 가지 기능 블록으로 구성된다. 금융망의 특성상 외부 기관 또는 개인 단말과 의 연계가 복잡하고 다양해서 유연성과 안정성이 동시에 요구된다.

둘째, 메시지 미들웨어(Message Middleware)는 J2EE 기반의 ESB 엔진을 사용하여 트랜잭션을 처리한다.

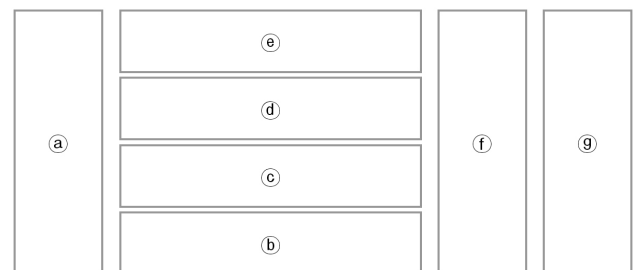
셋째, BPM 엔진은 오픈소스 BPM 엔진을 커스터마이징하여 ESB 엔진과 연동하고, 컴포넌트 들을 BPMN 2.0 노테이션에 맞게 태스크, 액티비티(Activity) 등의 컴포넌

트로 구성하였다.

넷째, 컴포넌트는 비즈니스 컴포넌트와 프레임워크 공통 컴포넌트 등의 용도에 맞게 4.1 컴포넌트 아키텍처에 서처럼 컴포넌트 저장소에 구성된다.

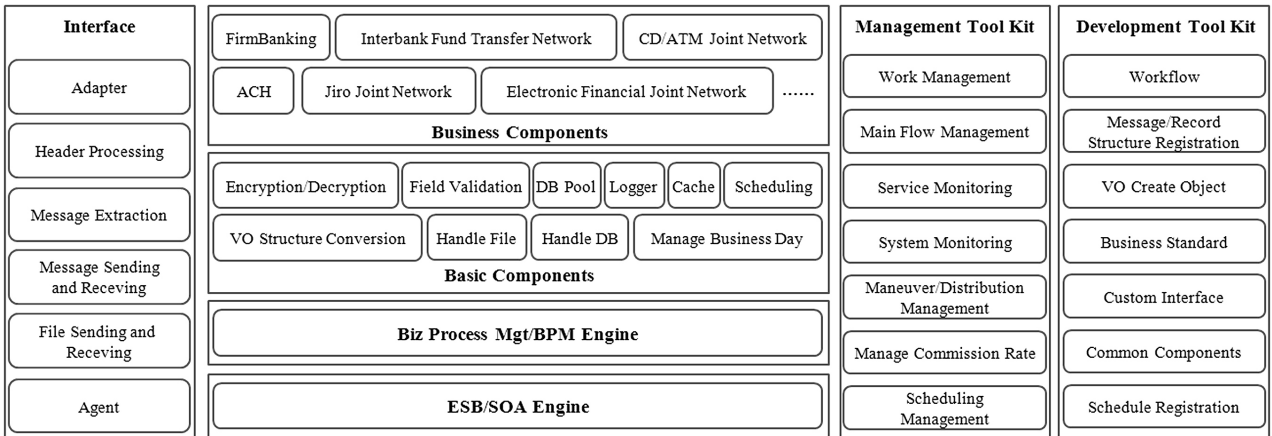
다섯째, 시스템 모니터링 등 일반 운용관리 업무 뿐만 아니라 서비스 동적 재구성을 위한 파라미터 설정 등의 제어 기능을 갖는다.

여섯째, 개발도구는 BPMN 기준에 맞게 설계된 사상을 컴포넌트 들을 BPM 엔진에 연동하여 프로세싱 하도록 한다. PSF의 전체 구성은 <Figure 7>과 같다.



- a : Interface Module
- b : Message Middleware
- c : BPM Engine
- d : Biz & Framework Common Component
- e : Biz Component
- f : Management Tool Kit
- g : Development Tool Kit

<Figure 6> PSF Structure

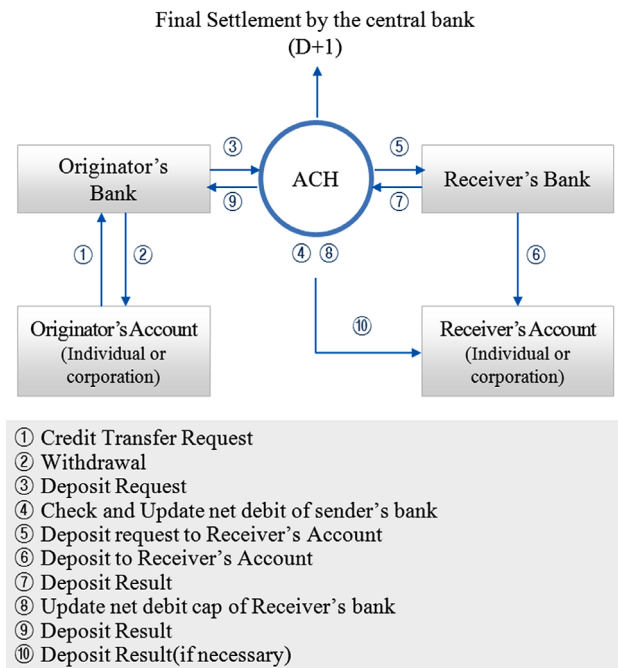


<Figure 7> PSF Architecture

3.5 결제청산 업무 적용 사례

결제청산(ACH)에서 계좌이체 영역에 대한 업무 프로세스 설계 내역은 <Figure 8>과 같다.

개인이나 기관의 계좌, 카드, 전자지갑 등으로부터 개인이나 기관의 계좌, 카드, 전자지갑 등으로 펀드의 이동으로, 개인의 입금이동은 단순한 펀드의 이동뿐만 아니라 세금납부, 정기구독지급, 용자 분할 납부와 같은 고지서 지급에도 사용할 수 있다. 대량 직접 계좌입금은 B2C, G2C로 다양화된 사업 모델에 따라서 서비스를 정의하는데 사용된다. 전형적인 서비스는 월급지급, 영수지급, 연금지급, 이자와 세금환급이 있다.



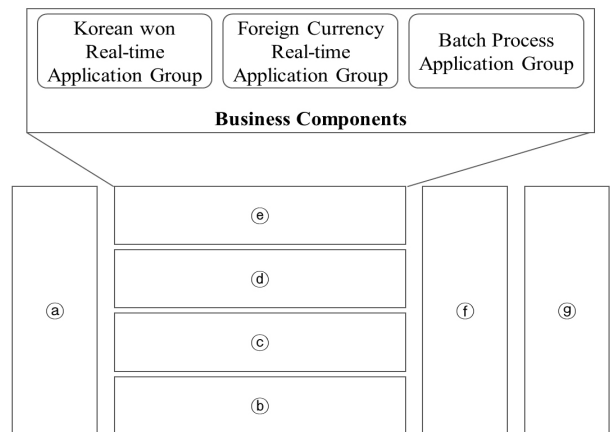
<Figure 8> Credit Transfer Process

4. 사례분석-적용 및 평가

4.1 베트남 W은행 펌뱅크시스템 구축 사례

펌뱅크 시스템은 은행에서 가입자 들을 대상으로 이 용기관 연계 및 파일 송수신 처리 서비스를 제공하는 기업 전용 전자금융업무 시스템으로 기업의 자금수납 및 자금업무를 간편하게 처리할 수 있도록 하며 효율적인 자금관리를 위한 시스템이다.

PSF 아키텍처에서 <Figure 9>처럼 비즈니스 컴포넌트로 펌뱅크에 특화된 원화실시간 처리, 외화 실시간 처리 일괄처리를 위한 컴포넌트 들을 사용하였고, 베트남의 상황에 맞게 베트남동과 미국달러 화폐 단위를 처리하도록 하였다. 관리도구(Management Tool)와 개발 도구(Development Tool)에서는 기본적으로 영문 메뉴를 제공하였고 비즈니스 요건에 맞는 UI 컴포넌트들을 사용하였다. 레거시 시스템과는 파라미터 설정으로 통신사양, 전문사양 등 연계인터페이스를 설정하였다.



<Figure 9> Firmbanking Construction Case

4.2 BOH(Bank of Hope) 사례

미국계은행인 BOH는 한국지사예 전자금융업무시스템을 구축하였고, 이를 위해 PSF에 전자금융을 위한 컴포넌트들을 사용하여 개발하였다. 전자금융업무는 금융결제원을 경로로 금융기관 간 실시간 온라인 타행거래 서비스를 위한 시스템이다.

BOH는 전자금융업무시스템 구축을 위해 <Figure 7>의 PSF 아키텍처에서 비즈니스 컴포넌트 영역에 전자금융을 위한 컴포넌트들을 사용하여 개발하였다. 사용된 컴포넌트는 타행이체, 수표조회, 수취계좌조회, 이체취소, 지급정지해지, 자금청구반환 등의 송금 업무와 순이체한도관리, 기관상태관리, 망상태관리 등의 관리업무 컴포넌트 등이다.

PSF에서 개발 도구(Development Tool), 인터페이스 모듈, ESB 엔진 그리고 BPM 엔진 등은 패키지를 사용하여 별도 수정 사항이 없었으며, 관리도구(Management Tool)에서 UI 부문은 사용자 요구 사항에 맞는 컴포넌트를 사용하였다. 추가적으로 기존에 없는 컴포넌트는 업무 요건에 맞게 부분적으로 신규 제작하였다.

4.3 평가

PSF를 사용하여 두 건의 지급결제시스템을 구축하였고 그 결과를 바탕으로 개선 내용을 확인한다. 비교대상은 국내 금융기관의 일반적인 지급결제시스템 구축 방식인 SI개발 방식과 비교하였다.

<Table 4> Quantitative Evaluation

	Domestic Construction Similar Case	Project A, B Case	Improvement Effect
Construction Period	6~10Months	3Months	More Than 50% Improvement
Invested Man Power	24~50M/M	6M/M	More Than 75% Improvement
Price Elasticity	Absolutely High Labor Cost	Low Labor Cost	Increase Price Elasticity

구축기간과 프로젝트에 투입되는 개발자 공수를 현저하게 줄임으로써 가격탄력을 높였다. 다만, 프레임워크를 구축하는 과정에서의 연구개발 비용과 추가적으로 발생하는 신규업무를 위한 컴포넌트 개발 비용은 개별 프로젝트의 원가에 고려하지 않아 평가에 한계가 있으나, 기 확보한 PSF를 기반으로 해외 프로젝트에 대응할 수 있는 기술경쟁력을 갖췄다고 평가한다.

<Table 5> Qualitative Evaluation

	Contents	Improvement Effect
Quality Consistency	Accurate Service Execution According to the Business Requirements Calculated at the Analysis Stage	Improvement and Distinctiveness
Stability	Reduces the Frequency of Errors After Opening the System	Relatively Short Stabilization Period
Maintenance	Use Visualization Tools to Increase Convenience by Sharing Business Requirement Design Screens on Operation System Screens	Long-Term Observation Required
Applying to New Business Requirements	Easier Process Change or New Addition According to BPMN2.0	Extend Dynamic Reconfigurable Range

5. 결론

한국 중소기업의 해외 시장 진출은 기업의 성장과 국가 산업발전을 위해 중요한 전략적 방향으로 제시되고 있다. 그러나 해외진출에 대한 장애요인, 특히 기술경쟁력에 대한 이슈가 제기되고 있다. 한국 중소 소프트웨어 기업의 기술 경쟁력 강화를 위해서 기업이 보유한 경험과 노하우를 자산화하여 축적하는 것이 필요하다. 따라서 본 연구에서는 소프트웨어 재사용 관점에서 프레임워크를 확보하는 것을 대안으로 제시하고 있다.

이를 위해서 첫째, 소프트웨어 재사용 관점에서 적용형 재사용과 반자동 개발이 가능한 프레임워크 설계의 요구사항을 선행연구를 분석하여 도출하였다. 둘째, 글로벌 지급결제청산 업무 개발을 위한 BPMN 2.0 기반의 프레임워크를 설계하였다. 마지막으로 jBPM을 기반으로 프로토타입을 개발하고 국내외 사례연구를 통해 평가 결과를 제시하였다.

향후 소프트웨어 재사용 관점에서 중소 소프트웨어 기업의 해외 시장 진출 방안에 대한 많은 연구가 진행될 것이다. 첫째, 다양한 국가, 산업에 프레임워크를 적용한 사례연구가 진행될 것이다. 둘째, 수평적 관점에서 공통적으로 사용될 수 있는 컴포넌트를 분류하고 일반화하기 위한 연구가 필요할 것이다. 마지막으로, 해외시장에서 소프트웨어 재사용 관점 프레임워크의 활용을 지원하기 위한 역량 개발 및 관리방안에 대한 연구가 필요할 것이다.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) Grant Funded by the Korea government (MSIP) (No.2017-0-006770011001, Global Intelligent E-Payment Framework).

References

- [1] Bassett, P.G., The theory and practice of adaptive reuse, 1997.
- [2] Frakes, W.B. and Kang, K., Software Reuse Research : Status and future, *IEEE transactions on Software Engineering*, 2005, Vol. 31, No. 7, pp. 529-536.
- [3] Jee, E.H. and Choi M.Y., ‘Software-driven society’ change as an industrial indicator, SPRi Issue Report, *Software Policy & Research Institute*, 2016, 2016-011.
- [4] Jee, E.H., A Research on the competitiveness of Software entrepreneurs in Korea, 2015.
- [5] Jee, E.H., Choi M.Y., and Yea, Y.S., 2017 Software Industry Outlook, SPRi Issue Report, *Software Policy & Research Institute*, 2017, 2016-017.
- [6] Jonson, R.E., Components, Frameworks, Patterns, 1997.
- [7] Kim, C.J., A Integration Research of Cloud Component based on PaaS for Enhancing Software Reusability, *Journal of the Korea Academia-Industrial cooperation Society*, 2013, Vol. 14, No. 2, pp. 868-877.
- [8] Kim, J.S and Oh, H.S., A Study on Implementation Method of Manufacturing Control Systems using Software Reuse, *Journal of Society of Korea Industrial and Systems Engineering*, 2008, Vol. 31, No. 4, pp. 168-176.
- [9] Kim, J.Y. and Kim, E.H., Market Characteristics & Global Strategy in SW industry, *Communications of the Korean Institute of Information Scientists and Engineer*, 2010, Vol. 28, No. 11, pp. 99-104.
- [10] Korea Financial Telecommunications & Clearings Institute, Statistic Analysis of Domestic and Foreign Settlement, 2014.
- [11] Laddaga, R. and Robertson, P., Self adaptive software : A position paper, SELF-STAR : International Workshop on Self-* *Properties in Complex Information Systems*, 2004, Vol. 31, p. 19.
- [12] Nam, Y.H., Kim, I.K., and Yoon, I.S., Study on International Bases for Improving International Competitiveness of Korean ICT Industries, *Journal of Information Technology and Architecture*, 2014, Vol. 11. No. 1, pp. 53-62
- [13] NIPA Software Engineering Center, Five for an effective reuse Software tip, 2014.
- [14] Oh, D.H., A Study on the Competitiveness of the Korean Software Industry, Samsung Economic Research Institute, 2011.
- [15] Park, K.H., A Cooperation Model of Small-Medium Software Enterprises Newly Penetrating into Emerging Markets, *The Korean Association of Small Business Studies*, 2013, Vol. 35, No. 4, pp. 343-375.
- [16] Park, K.H., Design of an Application Framework for Information Systems Development, 1998.
- [17] Ryu, S.I., Status and Recommendations of Software Industry, Issue & Trend, *DIGIECO*, 2014.
- [18] Self Adaptive Software, DARPA, BAA 98-12, Proposer Information Pamphlet, 1997.
- [19] The Bank of Korea, <http://www.bok.or.kr>.
- [20] Ye, E.S., Product-Line Architecture Development for Self-Adaptive Software, 2008.

ORCIDHuyn Ki Bae | <http://orcid.org/0000-0002-5556-0131>Yun Ji Ahn | <http://orcid.org/0000-0003-3771-3672>Kwang Ho Park | <http://orcid.org/0000-0001-9418-6026>

<Appendix>

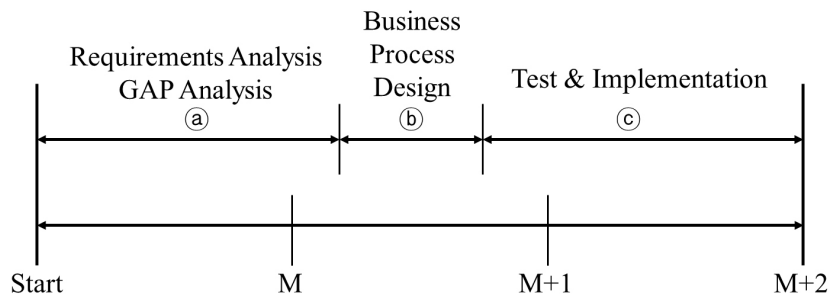
1. 베트남 W은행 펌뱅킹시스템 구축 개요

<Appendix Table 1> WBank Project Overview

Project Name	W Bank Vietnam Corporation Firm Banking Construction Business
Business Period	2017. 7. 01~2017. 09. 30
Invested Man Power	6M/M
Major Function	Company and User Management, Lookup Transfer History and Approval, Lookup Account History, Etc.

2. 펌뱅킹시스템 구축사업 특징

펌뱅킹시스템 구축 위하여 PSF를 사용하였으며 패키지 형태로 공급하여 3개월의 기간 동안 프로젝트를 완료하여 오픈하였다.



<Appendix Figure 1> Project Schedule

3. 펌뱅킹시스템 적용 화면

<Appendix Figure 2>는 기관으로부터 요청 받은 이체 거래 내역을 조회하는 화면 예이다.

View Transfer Details

Vendor Number	<input type="text"/>	Transfer Date	05-05-2017 ~ 11-12-2017	Business Identification Code	[1]Batch
Processing Status	Please select from the following Options				

No	Transfer Date	Vendor Number	Institution Name	Total Number of Req...	Currency...	Total Request Amount	Business Identification...	Location Identification	Processing Status
1	28-11-2017	100001235	Direa	10		100,000	Batch	Deposit	Batch Request File Receive Complete
2	22-11-2017	100001235	Direa	10		100,000	Batch	Deposit	Batch Request File Receive Complete
3	14-11-2017	100001235	Direa	5		50,000	Batch	Deposit	Processing Approval
4	17-11-2017	100001235	Direa	0		0	Batch	Deposit	Pending for Reporting Approval
5	31-10-2017	100001235	Direa	5		50,000	Batch	Deposit	Processing Approval
6	23-10-2017	100001235	Direa	21		2,333,331	Batch	Deposit	Batch Request File Receive Complete
7	24-10-2017	200000001	SAMSUNG CARD	0		0	Batch	Deposit	Process Complete
8	23-10-2017	100000002	Nice Payments	11		281,000	Batch	Deposit	Pending for Reporting Approval
9	23-10-2017	100001235	Direa	17		0	Batch	Withdrawal agreement	Batch Request File Receive Complete
10	19-10-2017	100001235	Direa	17		0	Batch	Withdrawal agreement	Process Complete

<Appendix Figure 2> View Transfer Details

4. BOH 전자금융업무 구축 개요

<Appendix Table 2> BOH Project Overview

Project Name	BOH Electronic Financial System Building
Business Period	2017. 06. 07~
Invested Man Power	6M/M
Major Function	Handing Business(Transaction from the Same Bank) Opening Business(Transaction from Other Bank) Bath Business(Transaction Details Summary Settlement)

5. 전자금융업무 구축사업 특징

BOH 전자금융업무시스템 구축을 위해서 SI개발 형태가 아닌 PSF를 이용한 패키지 공급형태로 프로젝트가 진행되었다. 패키지 공급 후 3개월 가량의 기간 동안 첫째, 겹(Gap)분석; 둘째, 프로세스 설계; 셋째, 실행모듈 생성 마지막으로 테스트를 실행하였다.

6. 적용 화면

<Appendix Figure 3>는 순이체 한도 상태를 조회하기 위한 화면이다.

The screenshot displays a web application interface for viewing net transfer limits. At the top, the title is "View net transfer limit". Below the title, there is a "Date of transaction" field set to "2017-09-18" and a "Logout" button. A warning message states: "※The inquiry request will result in the corresponding foreign transaction". The main content area is divided into two sections: "net transfer limit" and "Transaction general information".

net transfer limit

Center limit amount	1,000,000,000	Transfer limit Sojin-yul	50%
Transfer limit remaining amount	500,000,000	Maximum detection time occurrence	2017-07-12
Maximum Detected Amount	750,000,000		

Transaction general information

Request time	13:21:22	Response time	13:21:23	Response code	000
GUID	00000000000000000000000063003906			Last transaction time	13:15:32

<Appendix Figure 3> View Net Transfer Limit