

Contents

Introduction

사례 프로세스의 개요

관련 이론

여신심사프로세스의 문제해결과정

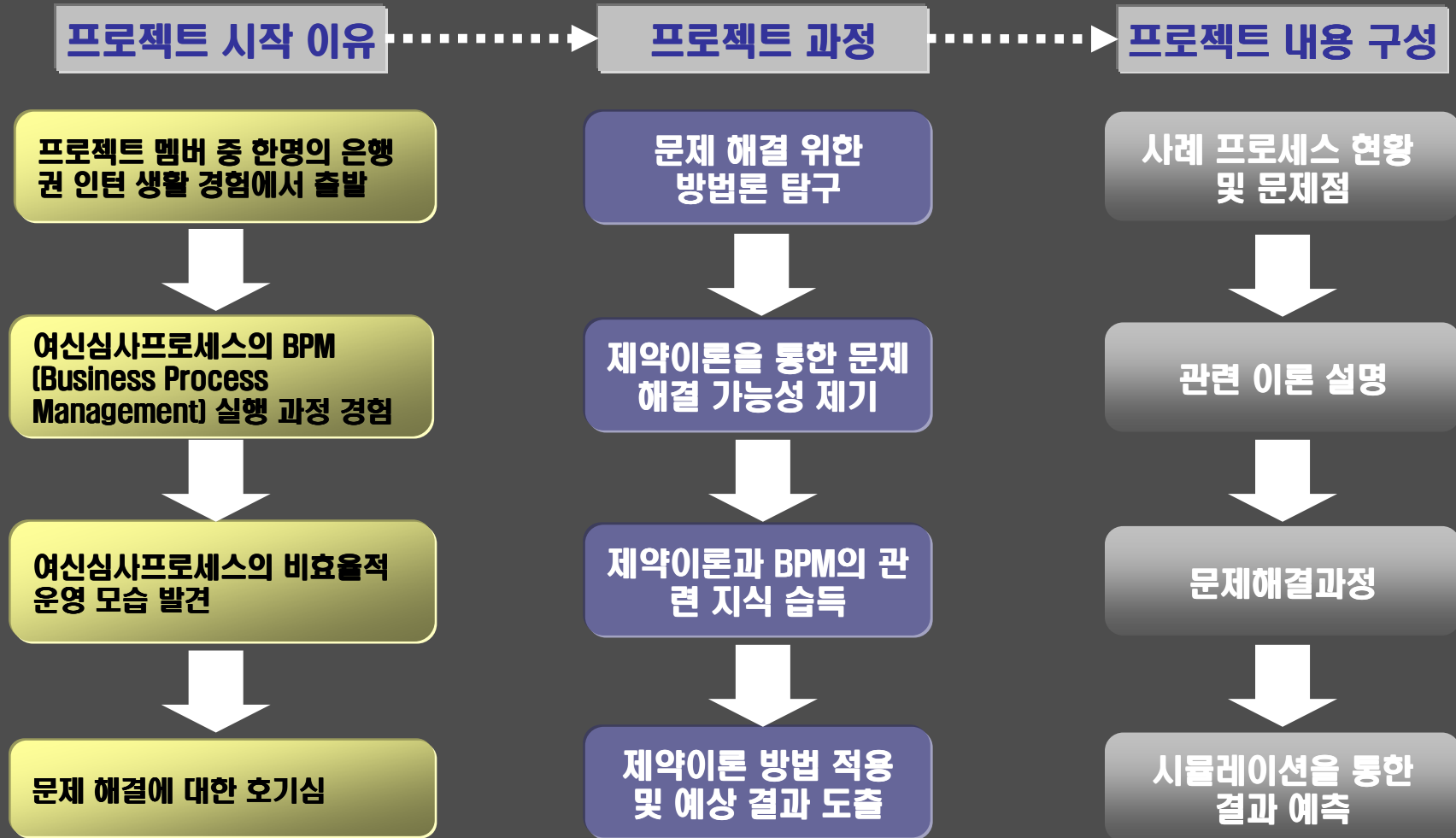
시뮬레이션 실험

결론

참고문헌

Introduction

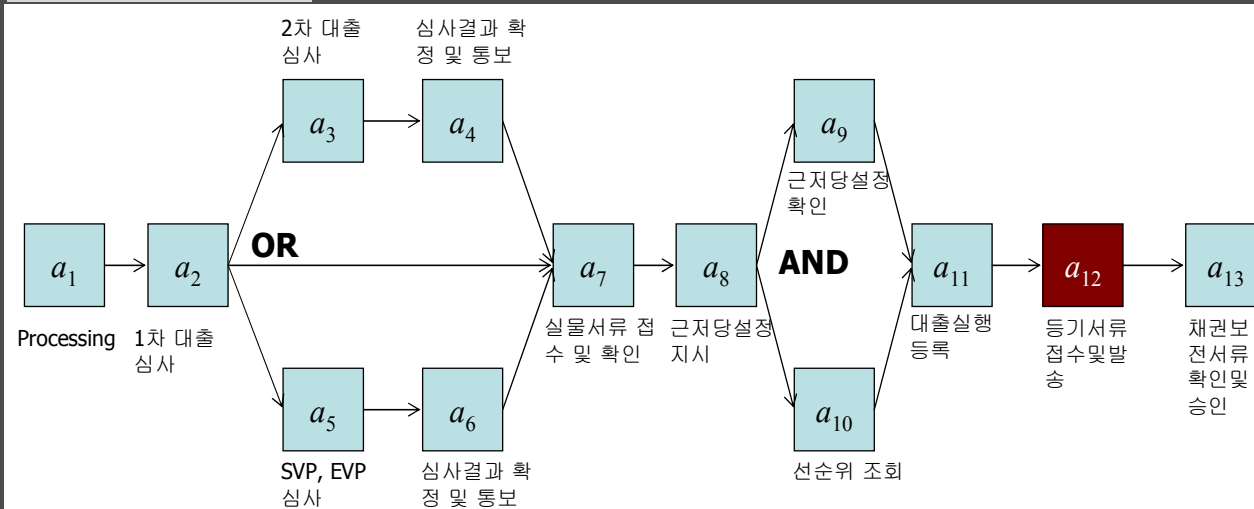
- 본 프로젝트의 개요를 설명한다.



사례 개요 [은행권 여신심사프로세스]

- 본론에 들어가기 앞서 문제해결의 대상이 되는 실제 은행권 프로세스의 현황을 소개한다. 그리고, 가지고 있는 문제점을 설명함으로써 본 프로젝트의 당위성을 보인다.

사례 프로세스



* 이 프로세스는 문제의 핵심만을 다루기 위해 실제 S은행의 프로세스를 단순화한 것임.

프로세스 현황

- S은행의 여신심사프로세스
- 각 지점별 심사 구조를 본사에서 직접 심사하는 구조(프로세스 표준화)로 전환하면서 상기 프로세스의 도입
- BPM을 통한 실행
- 단축효과: 10여일 → 2일로 단축
- 빠른 대출로 인해 수천억의 이자수의 발생

문제점

- a₁₂는 단순 문서 취합. 해당 업무 직전까지 처리된 관련 서류를 모두 취합해야 하는 업무.
- a₁₂가 완료되어야 a₁₃완료 후 대출이 이루어짐.
- a₁₂에서 과도한 업무부하로 지체 현상 발생.
- 단순 문서 업무인 만큼 추가적인 인력 투입 고려 안함
- 보다 빠른 여신처리를 통해 추가적인 이자수의 창출 여지가 보임

관련 이론 ; Business Process Management

- 본 프로젝트의 관련이론이 되는 BPM과 BPM상에서 설계되는 프로세스 모델에 대해 설명한다.

BPM (Business Process Management)

- 기업 내 프로세스를 정의, 설계, 실행, 개선하는 소프트웨어 시스템을 말함.

프로세스 모델

- BPM이 프로세스를 수행할 수 있도록 표현한 프로세스 형태
- 프로세스 단위업무와 선후관계를 포함하는 구조와 속성에 의해 표현

프로세스 모델은 다음을 만족하는 유방향 그래프 $P = (A, L, U, R)$

- $A = \{a_i \mid i = 1, \dots, N\}$: 단위업무의 집합. (a_i 은 i -번째 단위업무, N 은 P 의 단위업무 총개수)
- $L \subseteq \{(a_i, a_j) \mid a_i, a_j \in A \text{ and } i \neq j\}$: 링크 집합. (원소 (a_i, a_j) 는 a_i 가 a_j 보다 선행을 의미)
- $U = \{u_k \mid k = 1, \dots, K\}$: 업무참여자 집합. (K 는 전체 사용자의 수)
- $R \subseteq A \times U$: 역할집합. (특정 단위업무에 참여하는 업무참여자 역할)

관련 이론 ; Theory of Constraints (TOC)

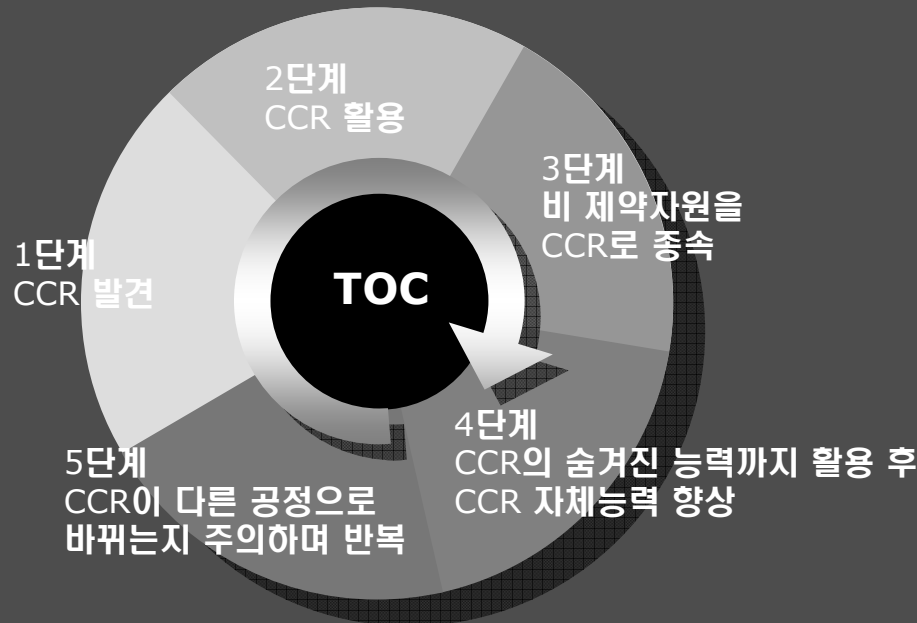
- 본 프로젝트의 관련이론이 되는 제약이론과 BPM에 적용할 제약이론의 방법론인 DBR을 설명한다.

제약이론 (TOC, Theory of Constraints)

- 시스템의 목적 달성을 저해하는 제약자원 (Constraint Capacity Resource, 이하 CCR)을 찾아 극복하는 경영기법
- 현금창출률 증가, 재고 감소, 운영비용 절감에 주력
- 제약이론의 절차

DBR (Drum-Buffer-Rope)

- 제약이론에서 효율적인 스케줄링 관리를 지원하는 방법론
- 효율적인 시간관리와 자원관리를 추구
 - 생산과정을 군대의 행진에 비유



- **Drum:** CCR의 업무처리속도. 전체 공정의 업무처리속도를 결정
- **Buffer:** 제약버퍼(Constraint Buffer)와 조립버퍼(Shipping Buffer)로 구성
- **Rope:** 제약자원과 시작공정 간의 통신장치. CCR의 속도에 맞춰 자재 투입속도 조절

관련 이론 ; 생산공정과 프로세스 모델의 차이

- 제약이론을 이용한 프로세스 실행방안을 설명하기 앞서 생산공정과 프로세스 모델의 차이점을 설명하고, 프로세스 모델 상에서의 제약자원을 정의한다.

| | 생산 공정 | 프로세스 모델 |
|---------|--|--|
| 프로세스 특징 | <ul style="list-style-type: none"> • 단위 공정 / 라우팅 고정 • 명시적인 자원 할당 | <ul style="list-style-type: none"> • 다양한 대안 실행 • 자원이 명시적으로 할당되지 않음 |
| Lot | <ul style="list-style-type: none"> • 로트 1이상 | <ul style="list-style-type: none"> • 프로세스 인스턴스를 로트 1로 간주 |
| 자원 | <ul style="list-style-type: none"> • 생산설비 • 공정 참여자 | <ul style="list-style-type: none"> • 업무담당자 • 업무 애플리케이션 |
| 업무 할당 | <ul style="list-style-type: none"> • 생산설비에 고정 • 그룹별 할당 | <ul style="list-style-type: none"> • 정의 : 1차 할당 • 실행 : 업무담당자에게 전달 |

그 외 프로세스 모델 특징

- 자원은 업무담당자에 한정함.
- 업무담당자는 동시에 다수 개의 업무 수행
- 같은 업무를 다른 업무담당자가 수행
- 업무담당자가 처리하는 단위업무가 지체되면, 할당된 또 다른 단위 업무들은 대기해야 함.

프로세스 모델의 "제약자원":

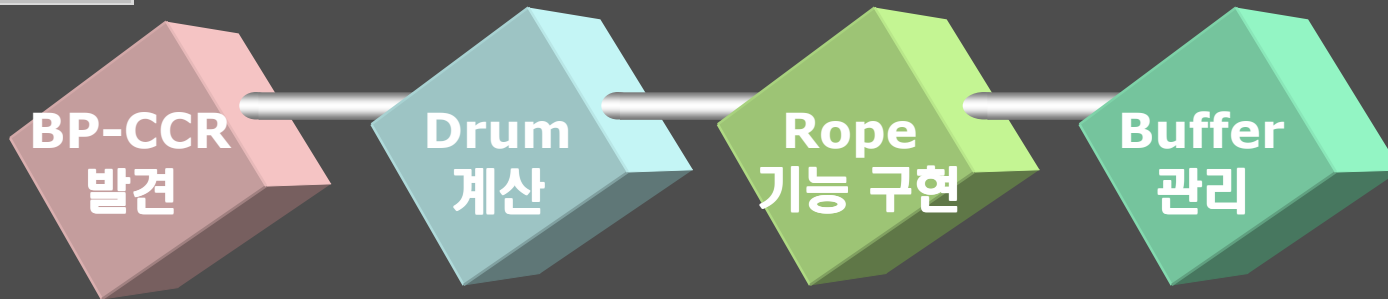
- 현 시점에서 업무 목록에 많은 단위 업무가 대기하고 있는 업무 담당자.
- 즉, 업무부하가 가장 큰 업무담당자.
- BP-CCR(Business Process-CCR)이라고 명명.

$$WL_{CCR} = \max \{WL_i \mid WL_i \text{는 } i\text{-번째 업무담당자의 업무부하, } i=1, \dots, I\} \text{일 때, BP-CCR은 업무부하가 } WL_{CCR} \text{ 이 되는 업무담당자}(u_{CCR}) \text{이다.}$$

여신심사프로세스 문제 해결 과정

- 관련 이론을 토대로 여신심사프로세스의 문제해결방법을 제시한다.

문제해결과정



- 1) 여신심사프로세스에서 부하가 가장 많이 걸리는 업무담당자를 확인한다.
- 2) 1)에서 구한 프로세스를 통해 Drum을 계산한다.
- 3) BPM에서 Rope를 구현시킨다.
- 4) BP-CCR의 업무가 중단되거나 과도하게 집중되지 않도록 Buffer를 관리한다.

여신심사프로세스에서 BP-CCR 발견 및 Drum 계산

- 여신심사프로세스에서 BP-CCR이 되는 업무담당자를 찾아내고 Drum을 계산한다.

BP-CCR 발견

- 평균적으로 심사에는 약 2일(16시간) 소요
- 등기서류접수 및 발송 프로세스(10여명의 심사자를 거쳐 처리된 결과를 각 지점으로 발송하는 곳)에서 업무담당자의 업무시간 약 2시간으로 가장 큼(문서량이 많고 내용이 복잡)

BP-CCR 발견

: 등기서류접수
및 발송 업무담당
자

Drum의 계산

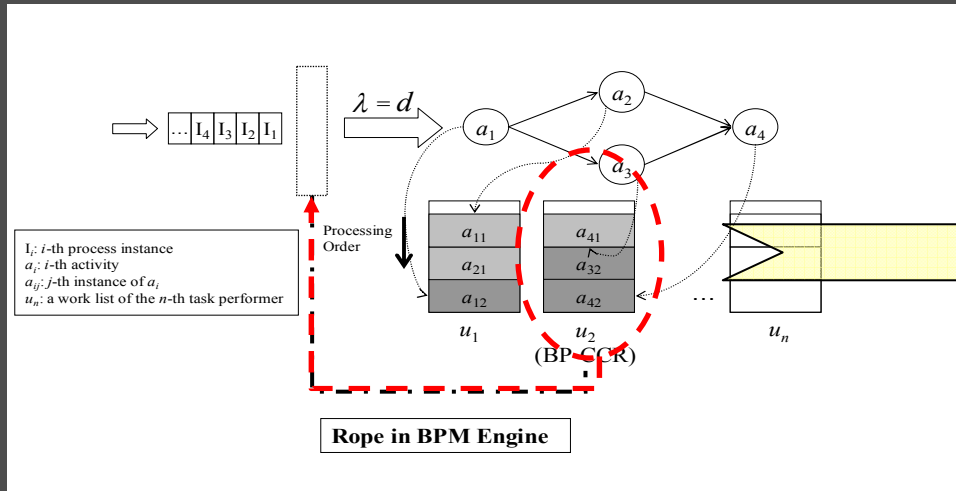
- 등기서류접수 및 발송 프로세스에 속한 담당자 2명
- BP-CCR 업무담당자의 해당업무 예상처리시간 (ET): 130분
- 업무담당자가 이 업무에 참여한 시간 T_{CCR} 동안
BP-CCR은 최대 $\frac{T_{CCR}}{ET} \times k$ 개 처리

$$\text{Drum } (d) = \frac{k}{ET}$$
$$= 1/65(\text{개/분})$$

여신심사프로세스에서 Rope 구현

- 여신심사프로세스에서 Rope를 구현한다.

Rope 구현



Rope의 목적

등기서류접수 및 발송 업무담당자의 업무 과부하 조정 및 업무 지연 방지

Drum = 1/65(개/분)이므로,
Rope = 65분 당 1개의 인스턴스 투입

- 제약자원과 공정 속도에 맞춰 시각 공정에 자재를 투입하는 속도 조절
- 업무량과 처리시간에 따라 투입속도 갱신
=> 지속적인 프로세스의 효율적 진행 지원
- BP-CCR(등기서류접수 및 발송 프로세스에 속한 담당자 2명)의 업무처리속도인 Drum과 동기화함.

Rope 구현

업무 투입의 지속적 갱신

↓
등기서류접수 및 발송 업무담당자가 업무를 지연시키는 것 방지

여신심사프로세스에서 Buffer 관리

- 여신심사프로세스에서 BP-CCR의 Buffer를 관리하는 방안을 설명한다.

Buffer 관리

- Buffer는 BP-CCR(등기서류접수 및 발송 프로세스에 속한 담당자 2명) 앞에 쌓이는 업무목록을 뜻함.
- TOC 에서 Buffer는 프로세스가 지속적으로 진행하되 재고를 최소화하고 운영비용을 절감하기 위해 존재함.
- 원활한 프로세스 운영을 위해 BP-CCR 앞에 시간과 비용의 관점에서 적절한 크기의 Buffer를 두어야 함.
- 프로세스 효율의 관점에서의 Buffer 연구가 미흡함.
- 프로세스 효율을 최적화하는 Buffer 크기는 상황과 관점에 따라 달리 접근하여 계산되는 성질을 지님.

Buffer의 목적

BP-CCR 앞의 업무 재고에 의한 비용을 최소화

Buffer 크기는 프로세스 환경에 맞도록 **경험적으로** 접근한다

Must

Buffer 관리

BP-CCR의 Buffer 크기를 다양하게 하여 BP-DBR의 효과 비교

여신심사프로세스 환경에서 가장 효과적인 Buffer 크기 결정

여신심사프로세스의 시뮬레이션 실험

- 지금까지 설명한 과정이 실제로 효과적인 방법인지 시뮬레이션 실험을 통해 보인다. 먼저, 모델을 어떻게 설계하였는지 설명하고, 실험 결과 일부를 분석한다.

How to Design



1. TOC를 이용한 여신프로세스의 효율성 개선 검증

| | 심사완료시간 (분) | 기한 내 심사승인건수 | 현재 진행중인 심사건수 |
|----------------|----------------|-------------|--------------|
| TOC 적용 | 446.888 (약 1일) | 140 | 6.351 |
| TOC 미적용 | 884.524 (약 2일) | 78 | 15.281 |

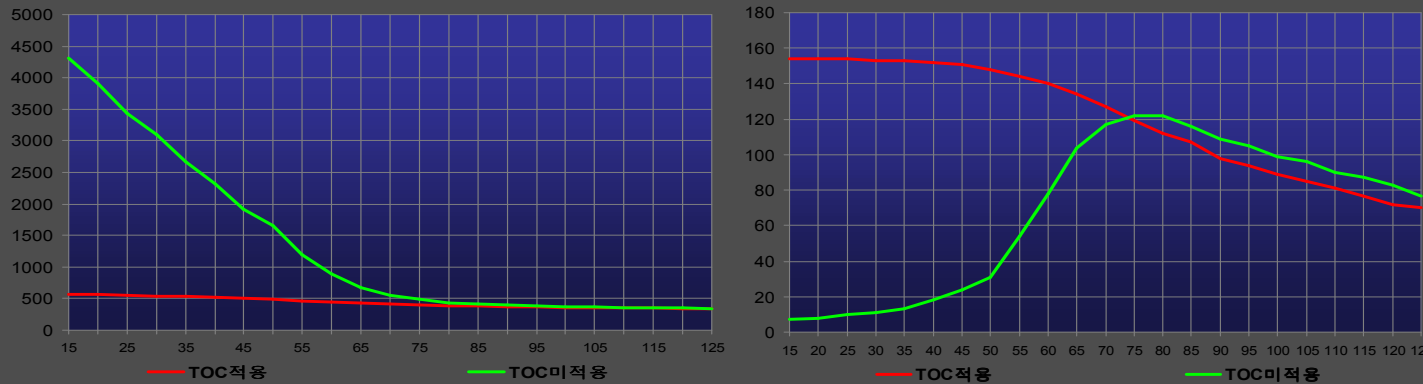
TOC을 반영한 프로세스와 반영하지 않은 프로세스간의 현저한 차이 존재

여신심사프로세스의 시뮬레이션 결과

- 이전 슬라이드에 이어 실험 결과를 분석한다.

2. TOC적용 시 여신신청빈도에 따른 효과 분석

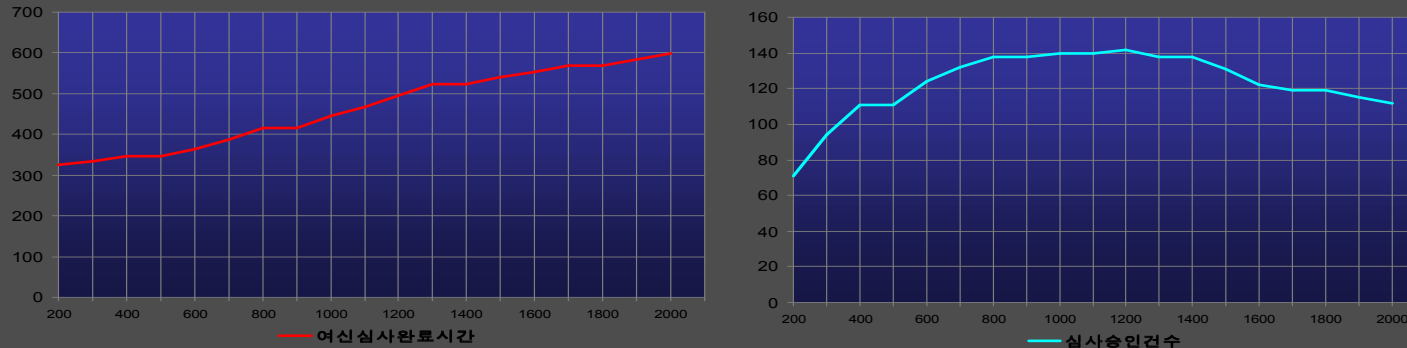
- BP-CCR의 처리시간은 130분이고 참여자가 2명이므로 드럼의 수치를 1/65(개/분)으로 시행



- 여신신청빈도가 빈번할수록 TOC효과증가
- 여신신청빈도가 Drum보다 더 빈번할 경우 TOC는 효과적임

3. 적정버퍼 크기의 결정(발생빈도 1/60(개/분))

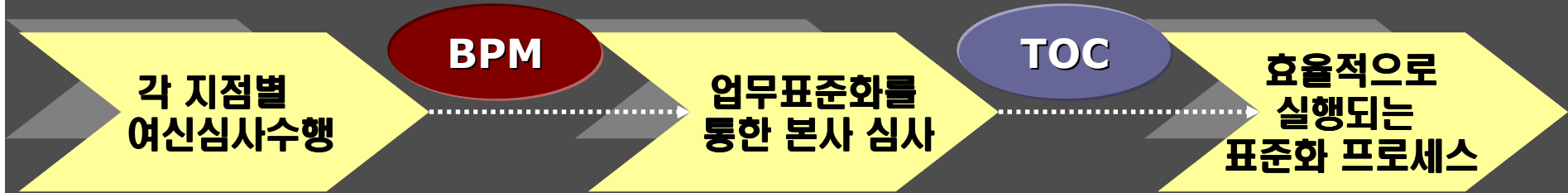
- 버퍼크기는 여신심사완료시간이 작더라도 여신승인건수가 적다면 효율성이 높다고 할 수 없음



- 약 1400의 버퍼크기에서 여신승인건수가 최대치를 가지며 이 이상일 경우 승인건수는 다시 감소
- 1400근처가 최적의 버퍼 크기

결론

- 본 프로젝트의 내용을 요약하고, 의미를 다시 한번 정리한다.



이미 BPM을 이용하여 상당한 금전적 이익을 창출했지만, 개선의 여지가 남아 있는 만큼 추가적인 경제적 이익을 기대할 수 있다.

TOC를 적용하면 ...

- 업무부하가 특정 자원에 편중
- 금융권과 같이 사람과 시스템을 통해 처리되어야 하는 프로세스



- BP-CCR을 집중 관리하여 업무부하를 분산
- 여신신청빈도가 잦을수록 이 방법론이 효과적
- BP-CCR의 버퍼 크기에 따라 프로세스 효율성 정도가 달라지며, 비교적 적합한 버퍼 크기가 존재함

참고 문헌

- **본 프로젝트를 위해 BPM과 TOC에 관련한 저명 서적 및 논문을 참고하였다.**
- **Chang, D., Son, J.H. and Kim, M.H., "Critical path identification in the context of a workflow", Information and Software Technology, Vol. 44, No. 7, 2002, pp. 405-417**
- **Goldratt, E. M. The Goal, North River Press, New York, 1992**
- **Hammer, M., The Agenda: What Every Business Must Do to Dominate the Decade, Three Rivers Press, 2003**
- **Kumar, A. and J. L. Zhao, Dynamic routing and operational controls in workflow management systems, Management Science, Vol. 45, 1999, pp. 253-272**
- **Wegner, P., "Why Interaction is more powerful than algorithms," Communications of the ACM, Vol. 40, No. 5, 1997**
- **Rhee, S. -H. et al, "Process-Oriented Development of Job Manual System," Lecture Notes in Computer Science, Vol. 3482, Springer-Verlag, Berlin, 2005, pp. 1259-1268**
- **Smith, H. and Finger, P., Business Process Management – The Third Wave, Meghan-Kiffer Press, 2003**