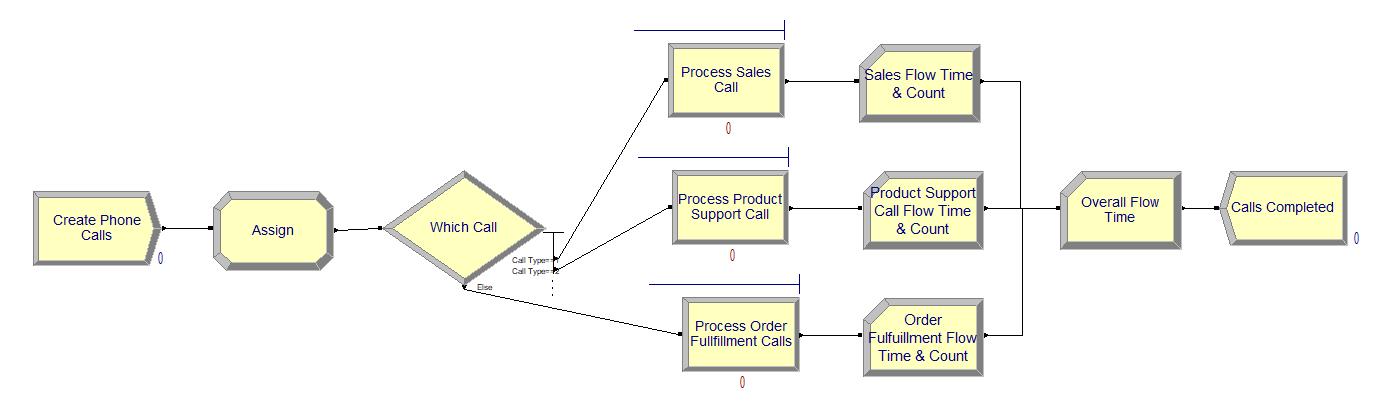
**<1>**



위와 같이 모델을 구성하면, 아래와 같은 Tally와 Counter set을 추가하여 각 통화 유형의 흐름 시간과 각 통화 유형의 수에 대한 통계를 수집할 수 있다.

테이블이(가) 표시된 사진

자동 생성된 설명 테이블이(가) 표시된 사진

자동 생성된 설명

10번의 복제를 기준으로 한 각 95% 신뢰 구간은 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

Sales Flow Time : 5.5521 0.35

Product Support Flow Time : 10.5154 2.59

Order Fulfillment Flow Time : 5.7269 0.53

Sales Count : 270 10.29

Product Support Count : 183.40 8.40

Order Fulfillment Count : 76.30 6.74

**<2>**

우선, 시간대별 Phone Call의 create를 다르게 하기 위해 Phone Call Schedule을 만든 후 이에 따른 도착으로 변경해준다.

또한 오전 11시부터 오후 1시까지 Product Support를 도와주는 알바를 고려하여 아래와 같은 Product Support Schedule을 만든 후 Resource의 Capacity를 변경해준다.

테이블이(가) 표시된 사진

자동 생성된 설명 텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명

이러한 고려 사항은 아래와 같은 결과를 얻게 한다.

테이블이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

참고로, 기존의 flow time, counter는 1번에서와 같고, utilization은 아래와 같다.

테이블이(가) 표시된 사진

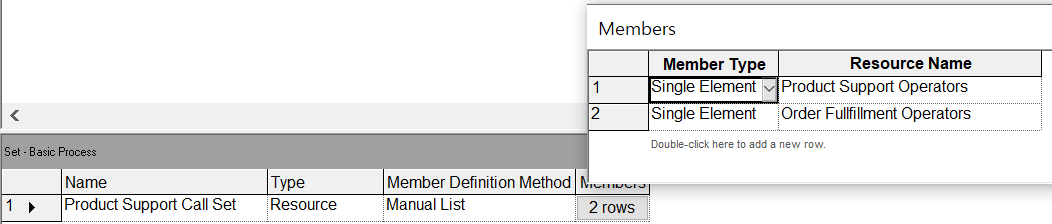
자동 생성된 설명

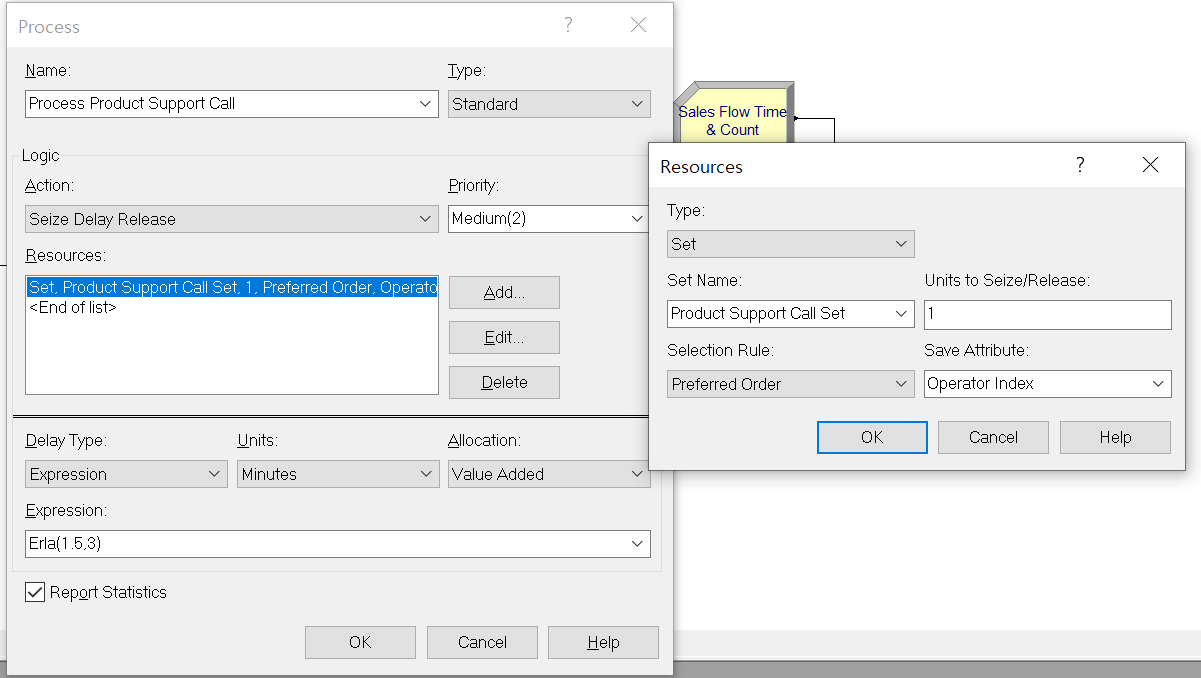
즉, 2번에서의 고려 사항은 신뢰구간이 겹치지 않는, 통계적으로 유의미한 Order Fulfillment Flow Time의 증가, Product Support Operators Utilization의 감소를 이끌어낸다고 할 수 있다.

**<3>**

order fulfillment operator는cross-trained되어 product support calls도 처리할 수 있기에, 아래와 같이 Set을 설정한 후, Product Support Call Process의 Resource를 알맞게 설정해주었다.

문제에서 제시한 우선순위를 고려하기 위해 Set에서 순서를 고려하여 Element를 추가한 후Selection Rule을 Preferred Order로 설정해주었다.





이 결과 flow time, counter, utilization을 살펴보면 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

기대되는, 예상되는 바와 상당 부분 일치하는 결과를 얻음을 확인할 수 있다.

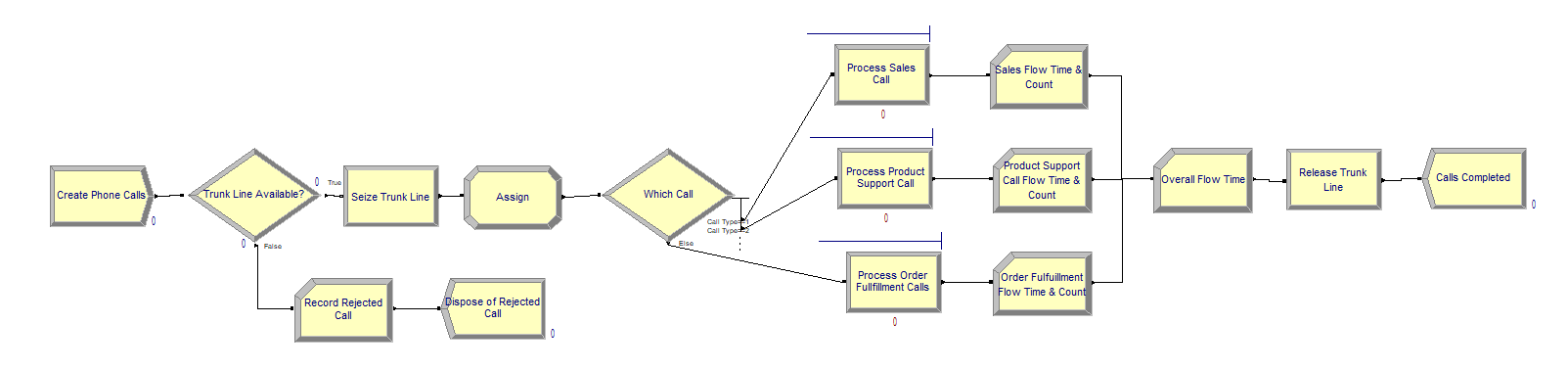
Product Support Flow Time은 크게 감소하고, Order Fulfillment Flow Time은 증가했다.

Product Support Count는 큰 차이가 없지만, Order Fulfillment Count는 감소했다.

Product Support Utilization은 감소했고, Order Fulfillment Utilization은 증가했다.

**<4>**

Trunk Line이라는 Resource를 만든 후, 문제 상황에 맞게 아래와 같이 모델을 수정했다.



전화가 걸려오면 Trunk Line을 하나씩 차지하고 마지막에 Release해주는데, 만약 남은 Trunk Line이 없다면 그 전화는 거절되고 counter를 이용해 그 개수를 기록한다.

결과는 아래와 같다.

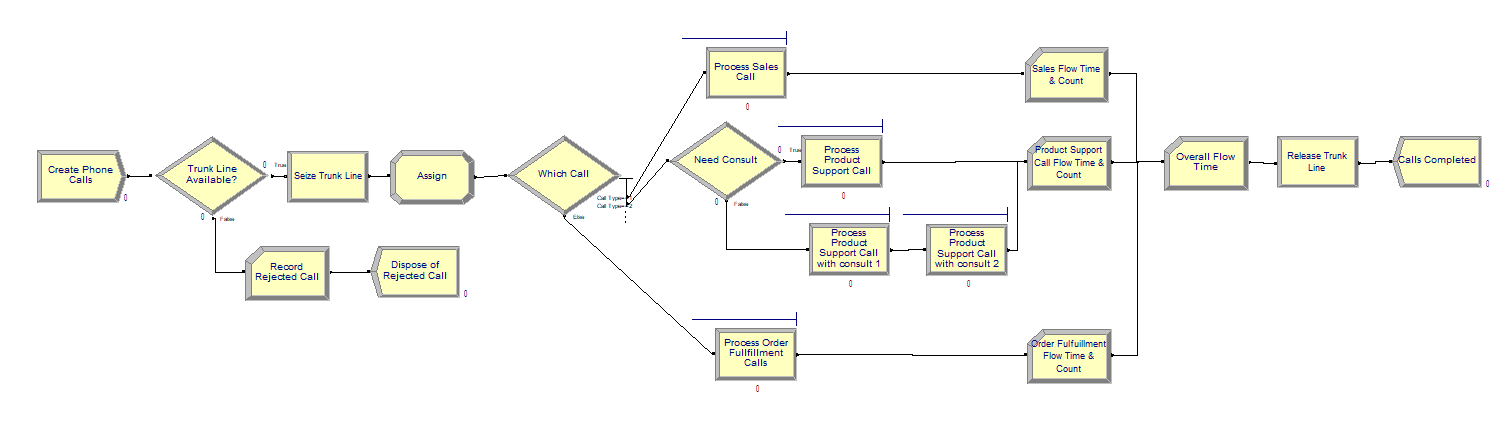
테이블이(가) 표시된 사진

자동 생성된 설명

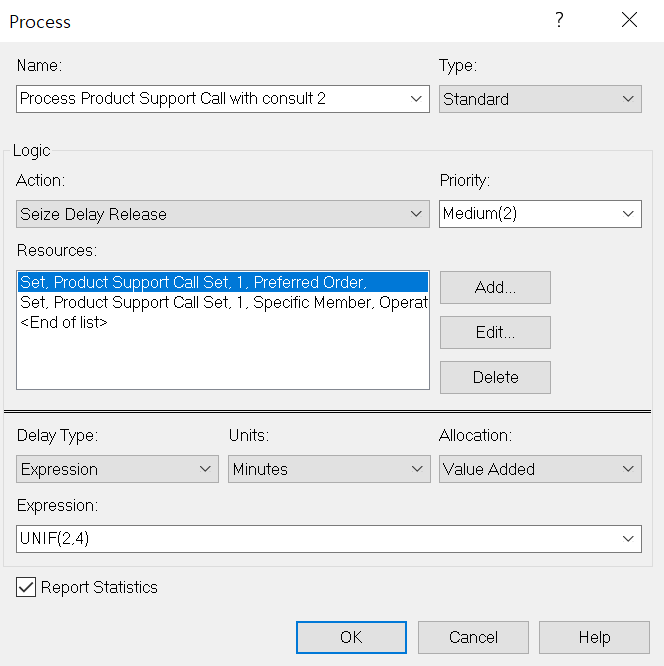
즉, 95% confidence interval on the number of lost calls per day는 5.6 3.74 이다.

**<5>**

10%의 Product Support Call을 상황에 맞게 처리하기 위해 아래와 같이 모델을 수정했다.



90%의 Product Support Call은 기존과 동일하게, 나머지 10%는 기존 process에 더하여 다른 상담원과 consult하는 process를 추가로 지정해주었다. 이 때 attribute를 저장하여 동일한 상담원을 지정할 수 있게 하였다. 추가로, Product Support 담당 중 ‘다른’ 사람과 상담한다고 했으므로 resource를 추가로 한 명 더 지정하였다.



이 결과 flow time, counter, utilization을 살펴보면 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

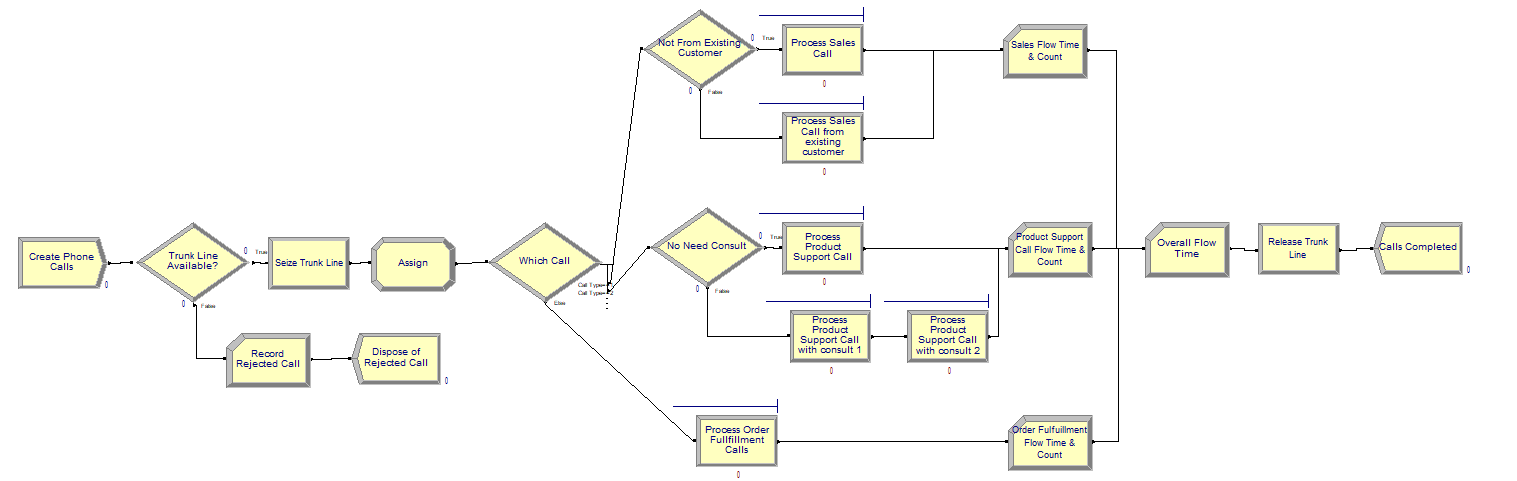
테이블이(가) 표시된 사진

자동 생성된 설명

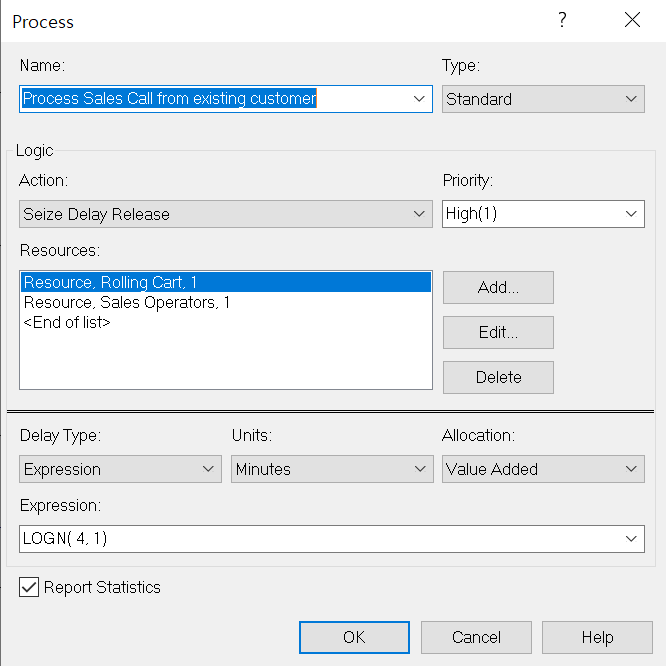
10%의 Product Support 과정에 작업이 추가되었으므로, 기대되는 바와 같이 Product Support의 Flow Time은 증가하고, count는 감소(유의미하지는 않음)하고, Utilization은 증가한 것을 확인할 수 있다.

**<6>**

아래와 같이 Sales Call에 대한 과정을 수정해주었다.



이를 위하여 우선 두 대의 Rolling Cart를 Resource로서 만들어준다. 그 후 기존 고객으로부터 전화가 온 경우(40%), Rolling Cart와 상담원 모두를 이용하며, 우선순위를 높게 부여하여 문제 상황을 만족시켰다.



이 결과 flow time, counter, utilization을 살펴보면 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

Sales Flow Time, Sales Count, Sales Operation의 Utilization이 크게 차이 나지 않는다는 것을 확인할 수 있다.

Rolling Cart라는 한정된 자원을 공유하기에 이로 인해 발생하는 대기가 Sales Process에 영향을 미칠 수 있지만, 그렇지 않았다. 그 이유를 분석해보면, Rolling Cart가 약 40%정도만 사용되며 이로부터 발생되는 추가 대기가 거의 없었음을 유추해볼 수 있다. 즉, 두 대의 Rolling Cart는 충분한 수량이다.