

# UML Components에 의한 컴포넌트 명세화 사례연구

안계중, 김태형, 이남용  
[ahnbe, thkim}@selab.soongsil.ac.kr,  
nylee@computing.soongsil.ac.kr

소프트웨어공학 연구실  
승실대학교 대학원 컴퓨터학과

## Contents

- Research Purpose
- Research Scope
- Research Method
- Component specification
- Findings & Conclusion

## Research Purpose

- 사례연구를 통하여 UML Component Design Process의 일관성 및 정확성을 검증한다.
  - ▶ Business Concept Model과 Use Case Model을 이용하여 Component Specification Process의 산출물인 Interface, Component Specification 및 Component Architecture 작성
- UML Notation을 확장해서 Component 명세에 적용하는 것이 올바른 것인지 확인한다.
  - ▶ <<core>>, <<type>>, <<data type>>, <<Business Interface>>, <<System Interface>>, <<Comp Spec>>으로 Component Specification이 충분한지를 확인
  - ▶ OCL을 이용하여 Component Specification 중 Component Specification, Interface Specification, Operation Specification, Constraint, Pre/Post Condition의 정의에 유용한지 확인

## Research Steps

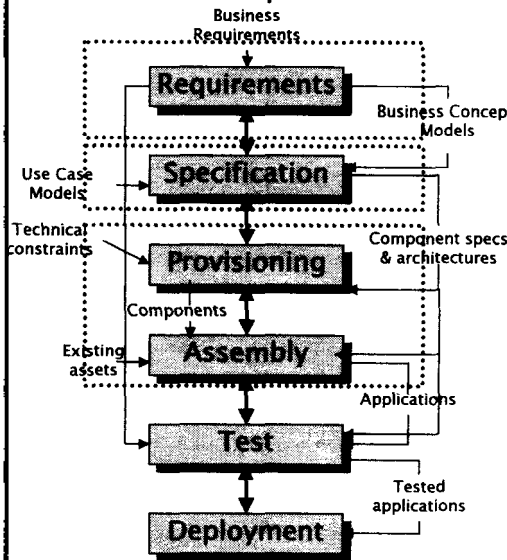
- UML Components에서 제시하는 프로세스 중 Specification의 세부 워크플로인 Component Identification, Component Interaction, Component specification을 수행한다.
  - ▶ Specification 단계에서 UML 기능을 확장하기 위하여 Stereotype를 사용
  - ▶ Component Spec, Interface, Operation의 Constraint, Pre/Post Condition 등을 기술하기 위하여 OCL을 사용

□ What is UML Components

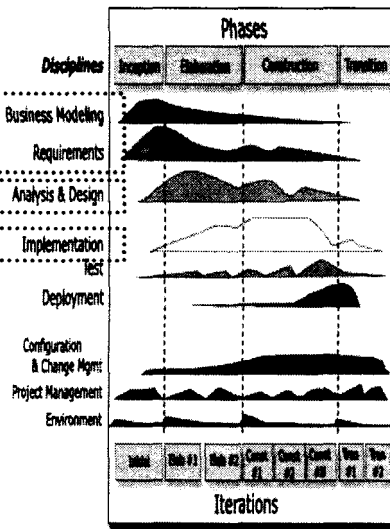
- ▶ UML Components는 컴포넌트 기반 소프트웨어 명세를 위한 프로세스
- ▶ 컴포넌트 시스템의 아키텍처와 컴포넌트 간 의존성을 명세화하는 표기법
- ▶ 표준 UML을 확장한 컴포넌트 모델링 기법

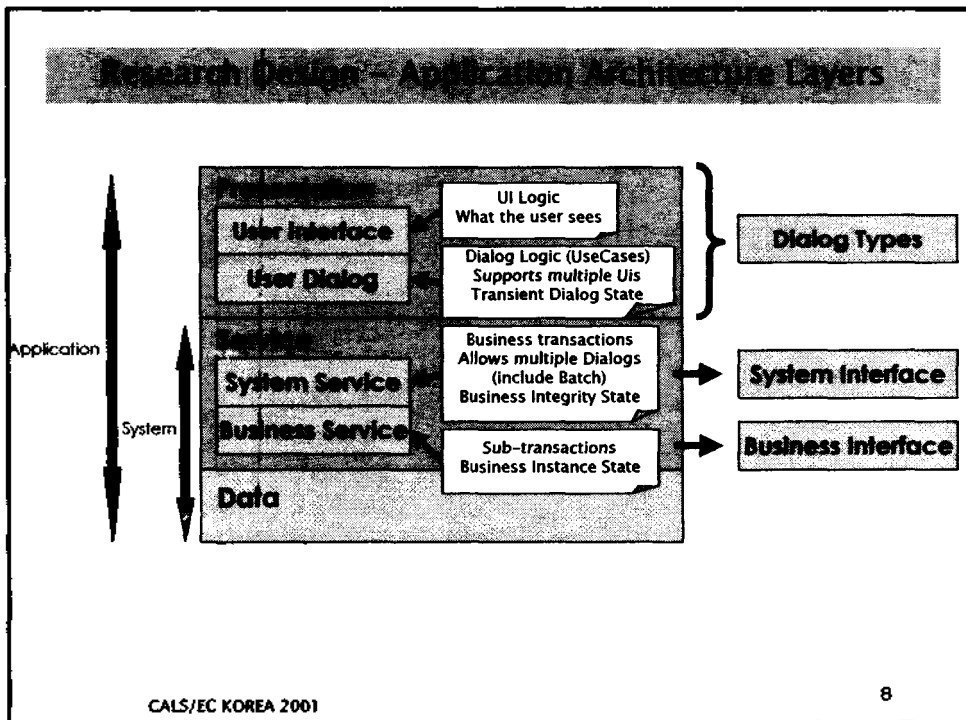
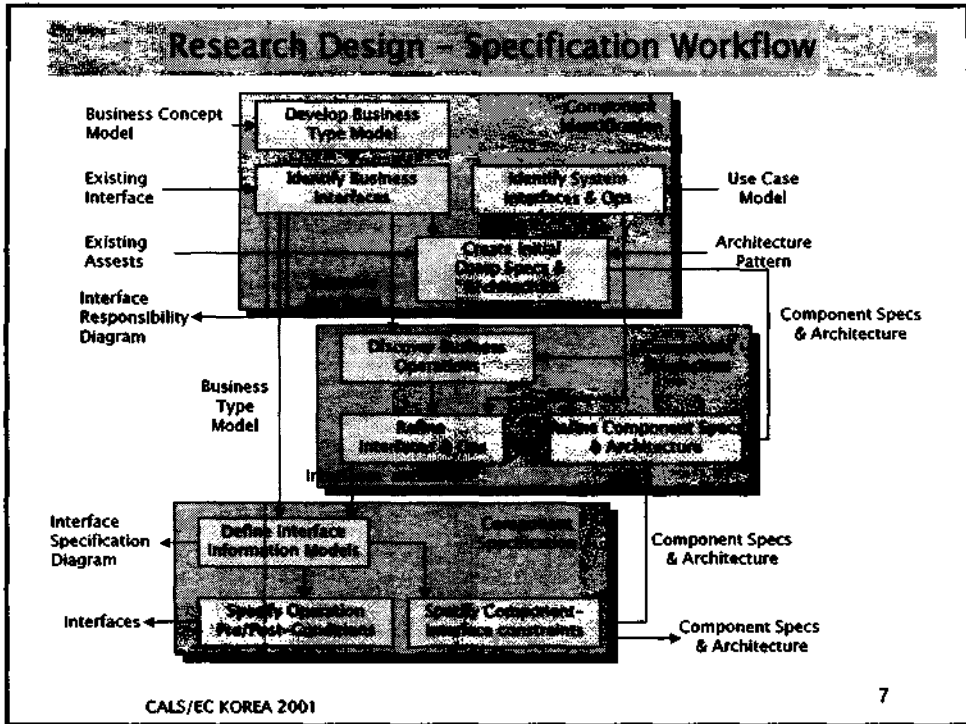
Research Design - Workflows in the Design Process

< UML Components >

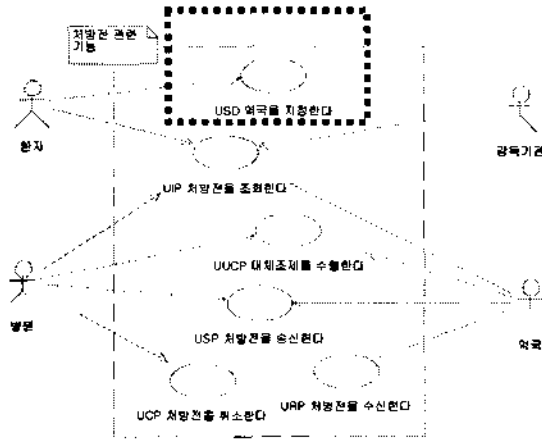


< RUP >





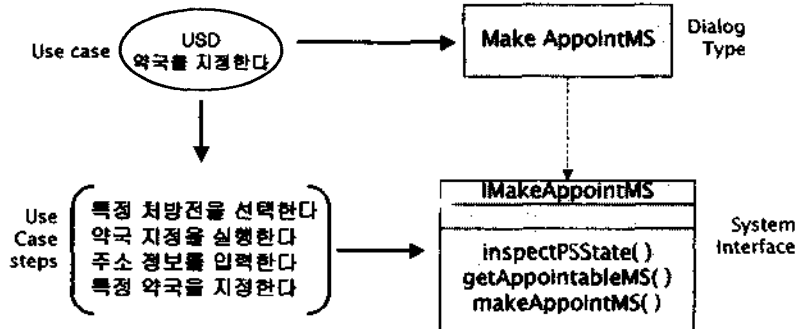
## Component Specification - Use Case Model



CALS/EC KOREA 2001

9

## Component Specification - Identify System Interfaces and Operations

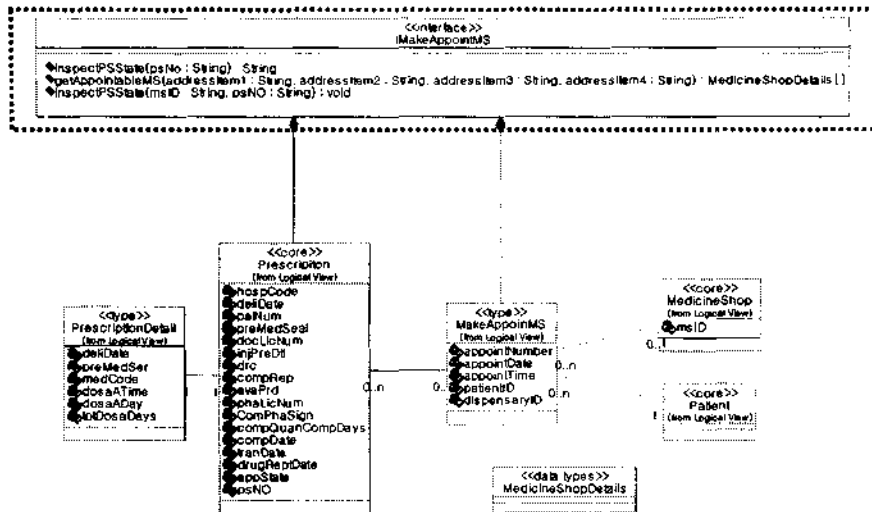


CALS/EC KOREA 2001

10



## Component Specification - Interface Specification Diagram



CALS/EC KOREA 2001

13

## Component Specification - Interface Constraint Condition

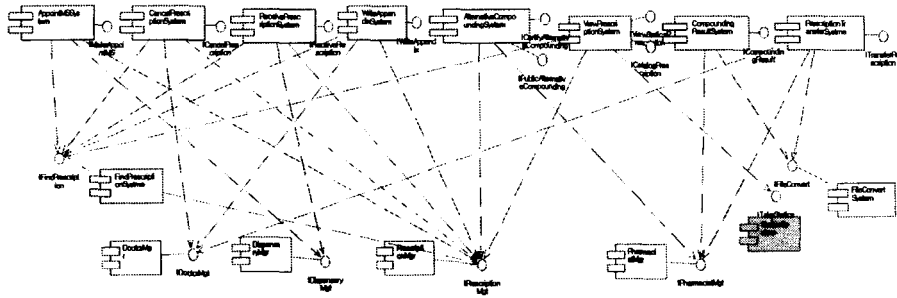
```

Context IMakeAppointMS : inspectPSState( in psNO : String ) : String
Pre :
    -- psNo는 유효한 처방전 번호(prescription number)이다.
    Prescription -> exists( p | p.number = psNo )
Post :
    -- 반환되어지는 처방전의 번호는 psNo와 일치해야 한다.
    thePrescription = prescription ->select(p|p.number = psNo) in
    -- specify the result
    result.patientName = thePrescription.patientName and
    result.dRegionCode = thePrescription.dRegionCode and
    result.licenceNumber = thePrescription.licenceNumber
Context IMakeAppoinMS : getAppoinableMS( in addressItem1:String,
in addressItem3:String, in addressItem4:String ) : MedicineShopDetails []
Pre :
    -- 인자값이 NULL이어서는 안된다.
    address.addressItem1.concat(addressItem2).concat(addressItem3).concat(addressItem4) <> NULL
Post :
    -- MedicineShopDetails의 주소정보는 입력 인자들과 일치해야 한다.
    result = medicineShopDetails->select(medicineShopDetails.address.includes(address) = true)
Context IMakeAppointMS : makeAppointMS( in msID : String, in psNO : String )
Pre :
    -- msID는 유효한 약국 ID이다.
    -- psNO는 유효한 처방전 번호이다.
    MedicineShop -> exists( m | m.id = msID )
    Prescription -> exists( p | p.number = psNO )
Post :
    --
    
```

CALS/EC KOREA 2001

14

## Component Specification – Component Architecture



## Component Specification

<b>Component Name</b>	<b>AppointMSSystem</b>
<b>Constraints</b>	<p>Context p : Prescription Inv:                  -- Prescription이 조제완료 혹은 약국지정 상태인 경우 또는 지정 가능 약국이 없는 경우에는 약국 지정 기능을 수행할 수 없다.</p> <p>p.claimd = p.appState -&gt; #1 or p.appState -&gt; #2 or                  MakeAppointMS.inspectPSSState.result = NULL</p>
<b>Offered Interface</b>	<b>IMakeAppoint</b>
<b>Used Interface</b>	<b>IFindPrescription, IDispensaryMgt, IPrescriptionMgt</b>
<b>Constraints between Interface</b>	<p>Context AppointMSSystem                  IMakeAppoint :: Prescription = IFindPrescription :: Prescription                  IMakeAppoint :: Dispensary = IDispensaryMgt :: Dispensary                  IMakeAppoin :: Patient = IPrescriptionMgt :: Patient</p>



## Findings & Conclusion

- UML Component Design Process를 이용하여 컴포넌트 명세화시 일관성 및 정확성을 확인함
  - ▷ Business Concept Model과 Use Case Model을 이용하여 Interface, Component Specification 및 Component Architecture 작성이 가능함
- UML Notation을 확장해서 Component 명세에 적용하는 효과적임
  - ▷ <<core>>, <<type>>, <<data type>>, <<Business Interface>>, <<System Interface>>, <<Comp Spec>>으로 Component Specification이 충분하였음
  - ▷ OCL을 이용하여 Component Specification, Interface Specification, Operation Specification, Constraint, Pre/Post Condition의 정의에 유용함
- ◆ UML, OCL과 같은 표준적인 방법을 사용하여 컴포넌트의 개발과 배포, 조립에 있어서 모든 역할 담당자들의 의사소통이 원활함
- ◆ UML Notation의 확장으로 새로운 종류의 구성요소를 생성하여 명세의 정확성을 높이지만 다른 개발 프로세스와 공유할 수 없는 단점을 가짐
- ◆ 컴포넌트 식별 방법이 공학적이기보다는 컴포넌트 설계자의 경험과 직관에 의존함
- ◆ UML Components는 기존의 객체지향 방법론과는 분명한 차이점을 가진다. 때문에 UML Components의 도입은 프로젝트 초기에 고려되어야 함

## References

- John Cheesman, John Daniels, *UML Components*, Addison-Wesley Publishing, 2000
- STERLING SOFTWARE, *CS/3.0, CBD96*, STERLING SOFTWARE, 1999