



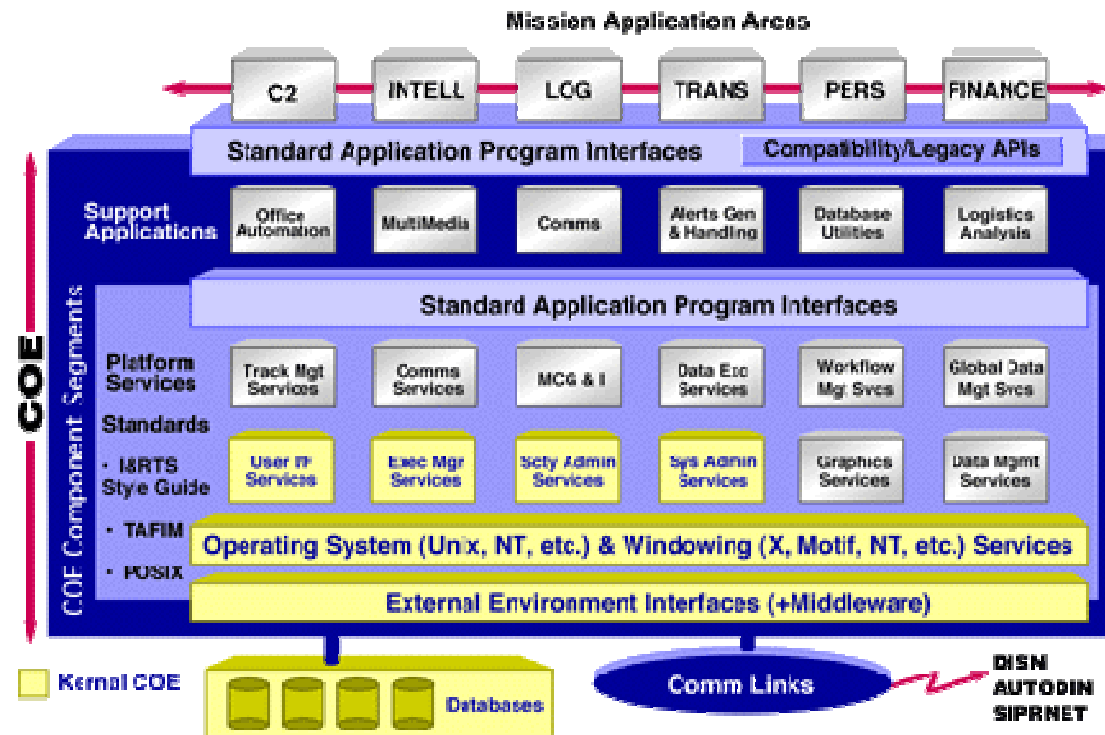
이종 임베디드 소프트웨어를 위한 개발도구 소개

홍익대학교 소프트웨어공학 연구실
컴퓨터 정보통신공학과 김영철 교수
bob@hongik.ac.kr

COE(Common Operating Environment)

❖ 문제점

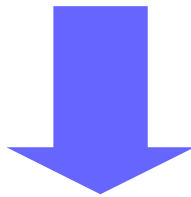
- 플랫폼 별 세그먼트 존재
- 세그먼트의 재사용이 어려움



기존의 개발 방법

❖ 기존의 임베디드 소프트웨어 개발

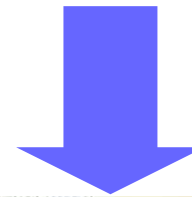
- 서로 다른 제품 개발시 각각의 S/W를 따로 개발
- 각 시스템마다 개발 소요기간 및 투입 인원이 중복 소요됨으로 비용이 증대



Windows Platform



Linux Platform



Unix Platform

e-MDA(Model Driven Archeture) 기반 방법

- ❖ 하나의 메타 모델링(Target Independent Model)을 통하여 각각의 다른 도메인에 맞는 타겟 종속적 모델(Target Specific Model)들을 만들고, 그에 따른 소스 코드(Target Dependent Code)를 개발하는 방법



기존 임베디드 SW 개발 비교

EMMA (Product Line) [Yang, Kang]	Harmony Process (MDD) [Douglass]	HiMEM (MDA/MDD Paradigm) [Kim]
1) 한번의 개발 라이프 사이클 후에, 핵심 자산 재사용하는 기법 2) 단종 의 임베디드 시스템을 적시에 경제적으로 개발 3) Feature Driven 4) 커스터마이징 용이 5) UML 2.0 적용	1) 한번의 개발 라이프사이클 중에, 정제된 프로토타입 재사용하는 기법 2) 단종 의 임베디드 시스템을 적시에 경제적으로 개발 3) 시스템 / 소프트웨어 책임 구분 4) UML 2.0 + SysML 적용 5) 코드 자동 생성기 필수	1) 한번의 개발 라이프사이클 중에, 하나의 메타 모델 을 재사용하는 기법 2) 이종의 임베디드 시스템을 적시에 경제적으로 개발 3) xUML + UML 2.0 적용 4) 커스터마이징 용이 5) 코드 자동 생성기 필수

적용사례

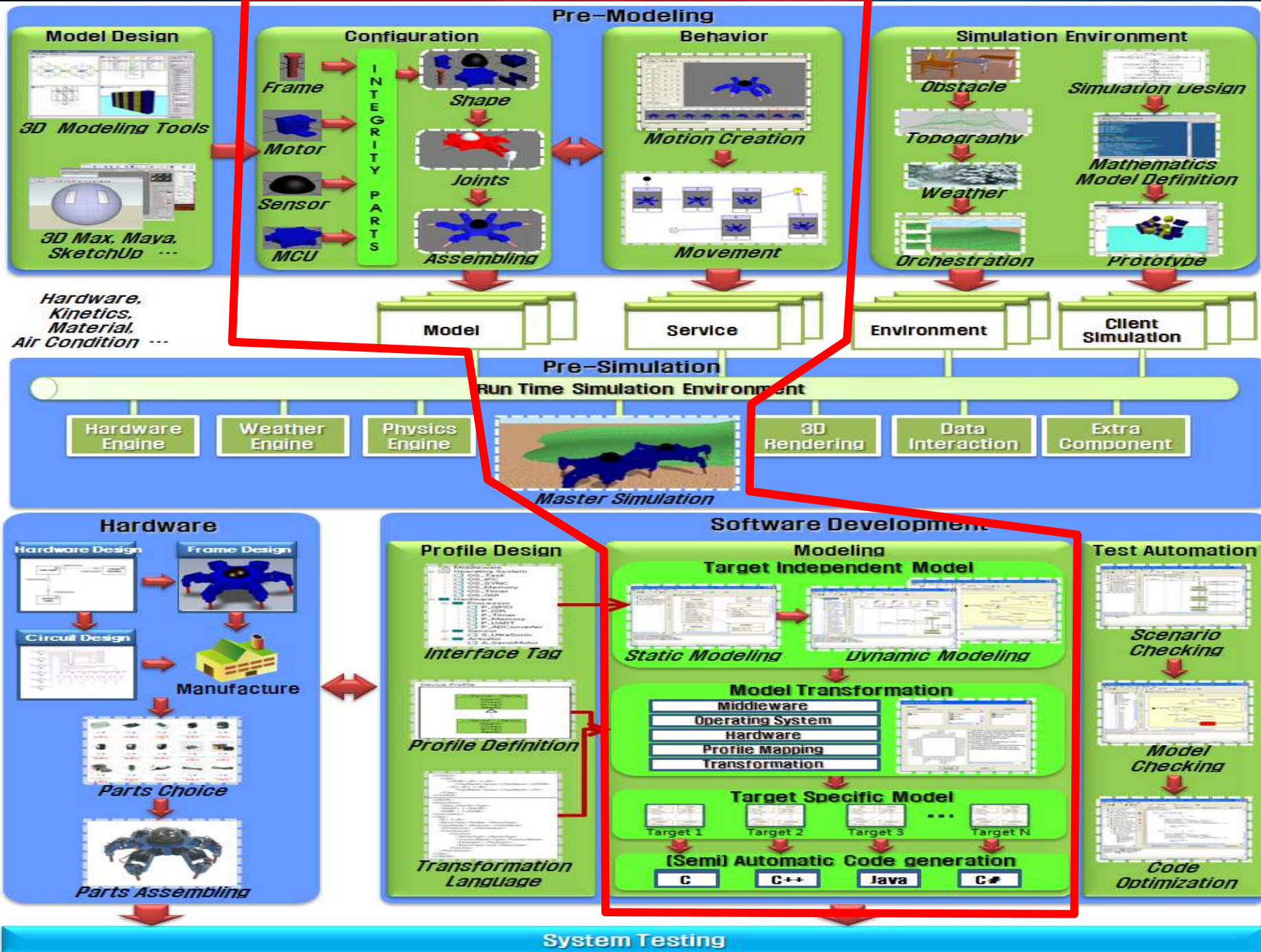


(a) SUGV 1

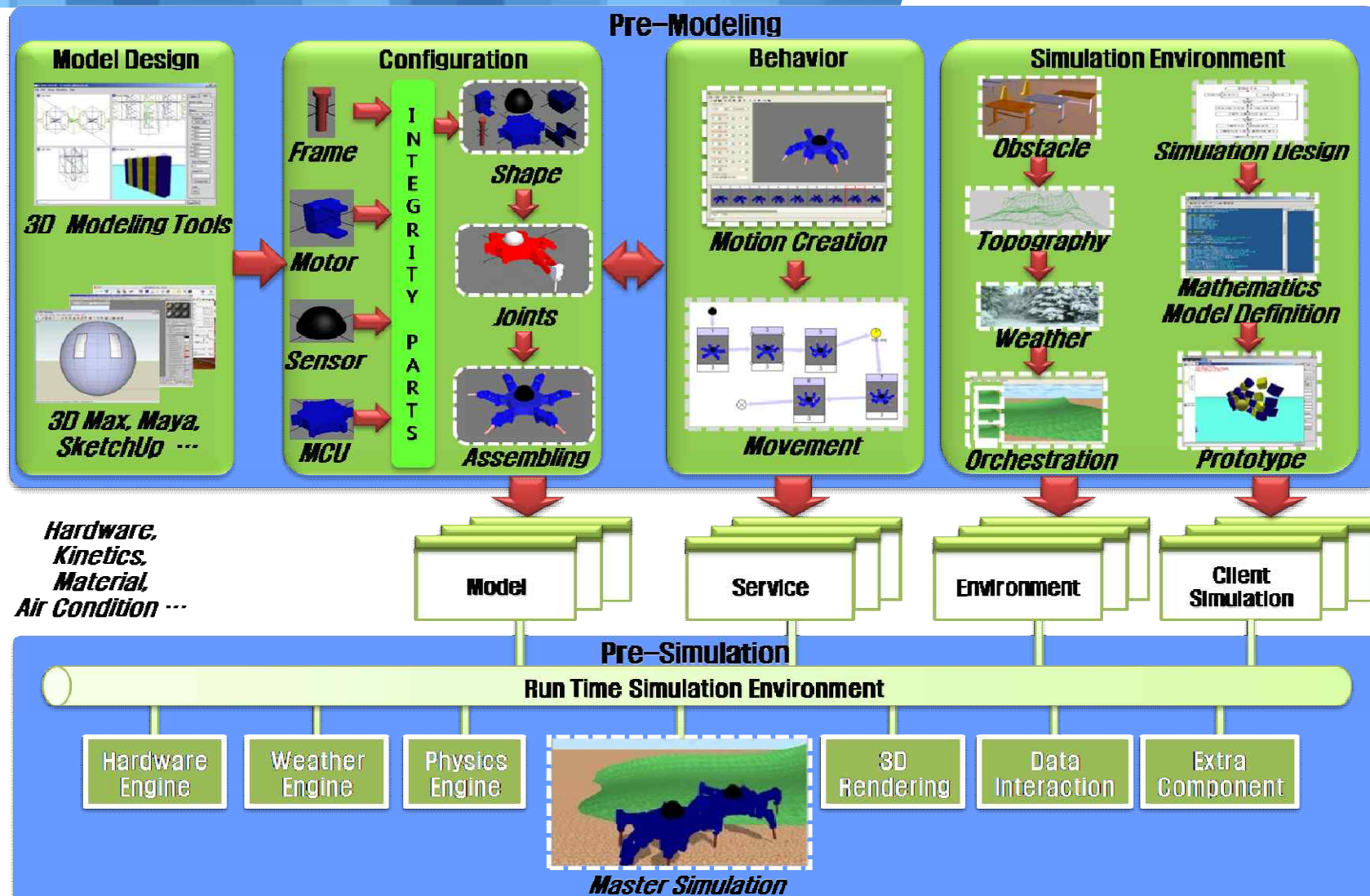


(b) SUGV2

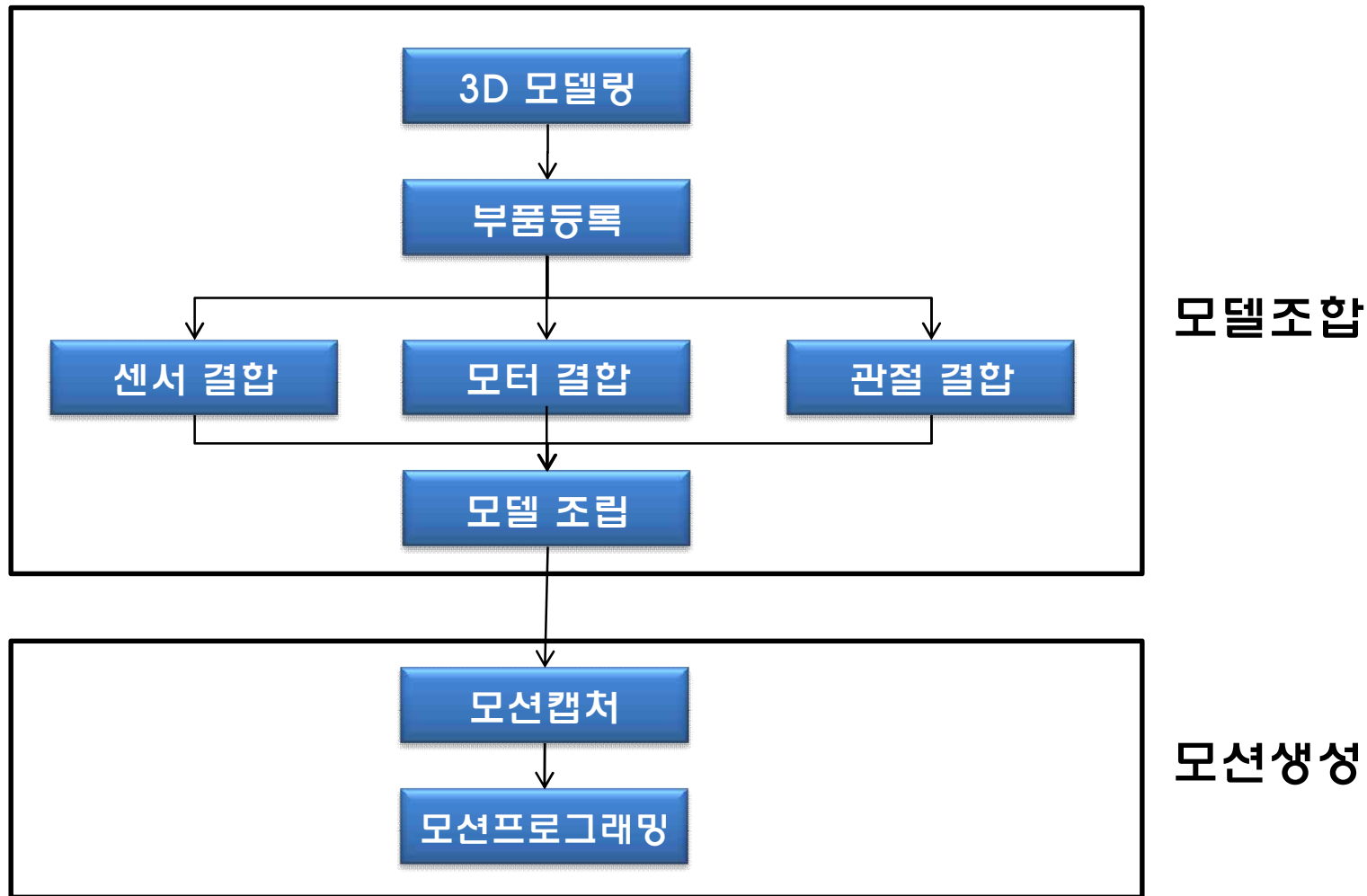
구분	(a) SUGV1	(b) SUGV2
Microcontroller	Ubicom SX48AC 20MHz	Atmel Atmega128 16MHz
OS	Javeline	brickOS
RAM	32 KByte	4KByte
EEPROM	32 KByte	128KByte
Sensor	2	2
Communication	Bluetooth	Bluetooth
Motors	18	18
JVM	On Hardware	No
Languages	Java	C/C++



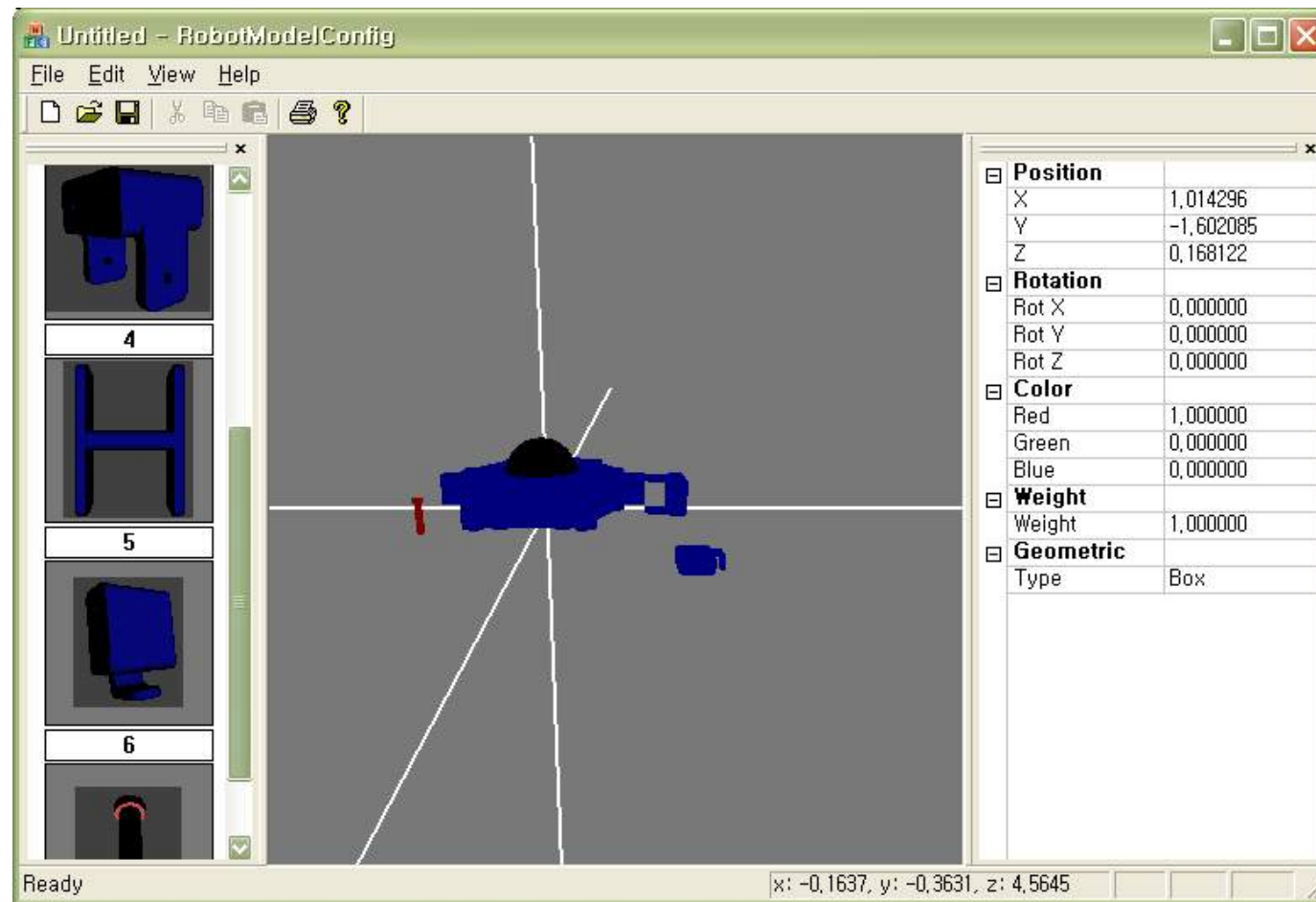
Pre-Modeling & Pre-Simulation



프리 모델링(Pre-Modeling) 절차



모델 조합 도구



모션 생성 도구

KMC Robotics for RealTime Modeling & Simulation © SE Lab., Hongik Univ. - 38)발장난.txt

File Edit View Tool Draw Tool Help

COM PORT
COM1 Connection

M01/M02/M03
66 83 90

M04/M05/M06
51 29 95

M07/M08/M09
91 94 145

M10/M11/M12
91 83 35

M13/M14/M15
146 149 86

M16/M17/M18
112 57 54

SPEED/DELAY
5 1

CONNECTION
이곳에 설명을 넣으세요.

3D Modeling Motion Based Language

2 5 7
3 3 5

13 12
5 5

1000 ms

5

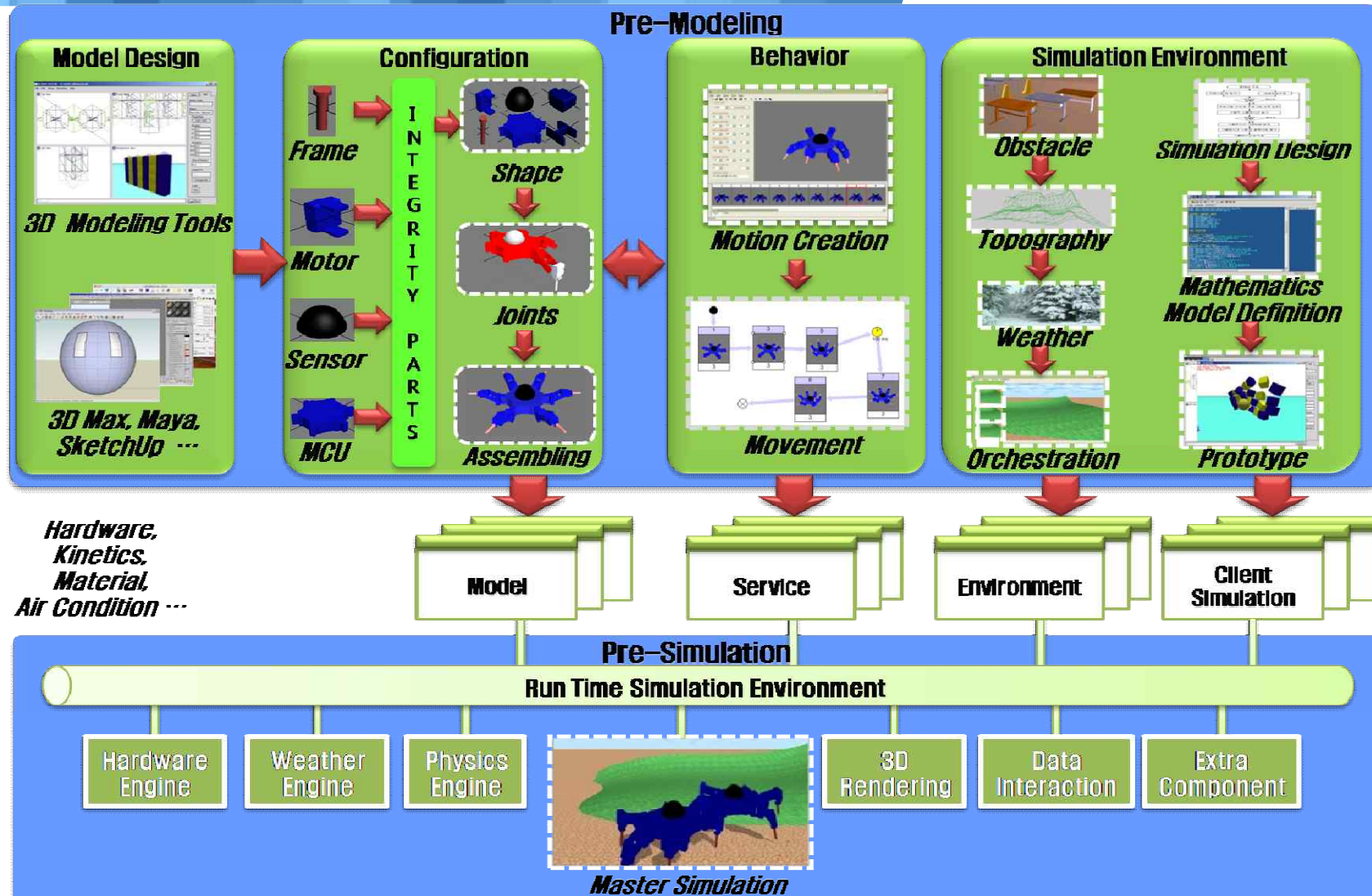
1 2 3 4

5 6 7 8 9 10 11 12 13

View Table

Ready

Pre-Modeling & Pre-Simulation

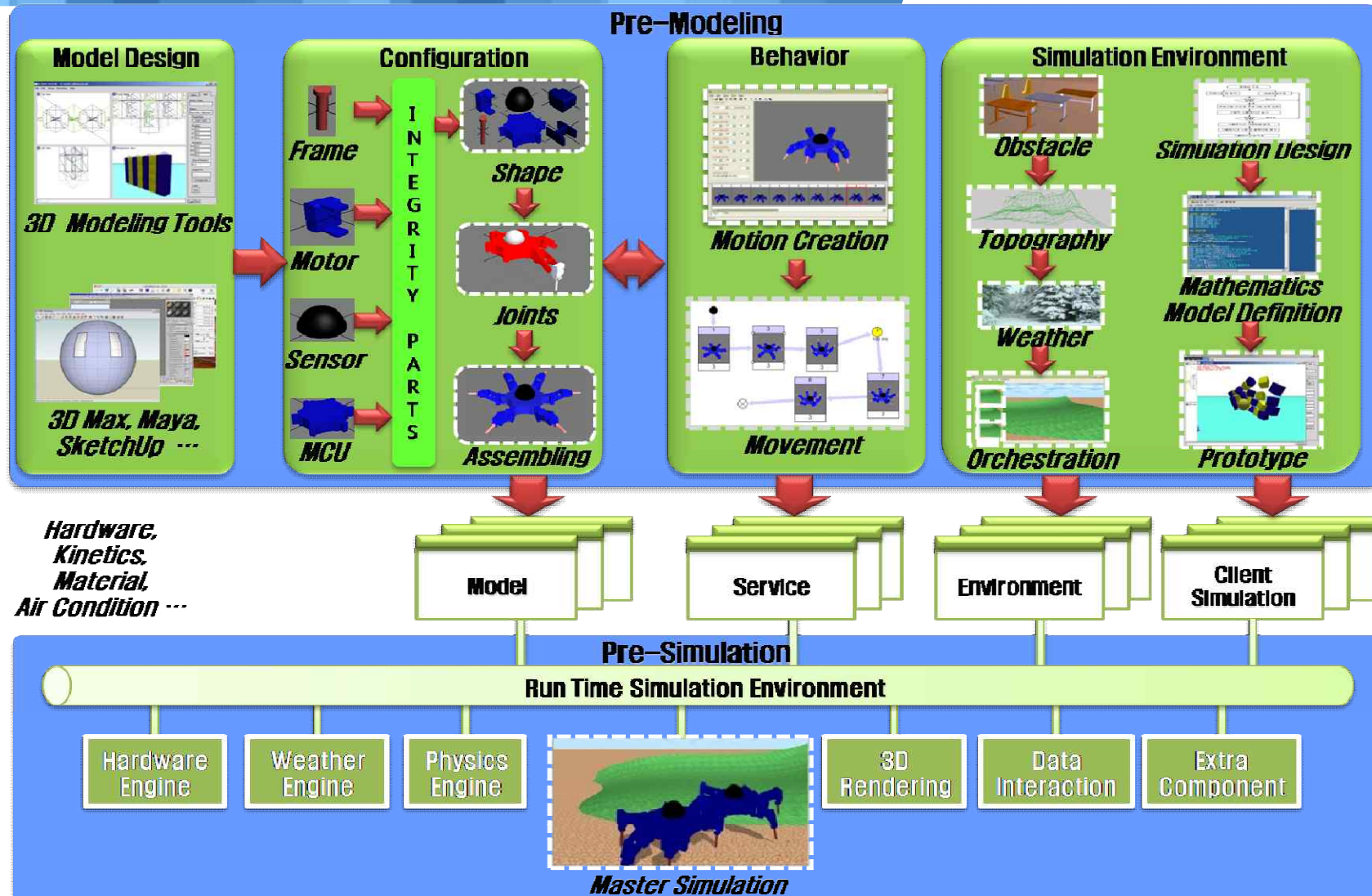


Simulation Enviroment



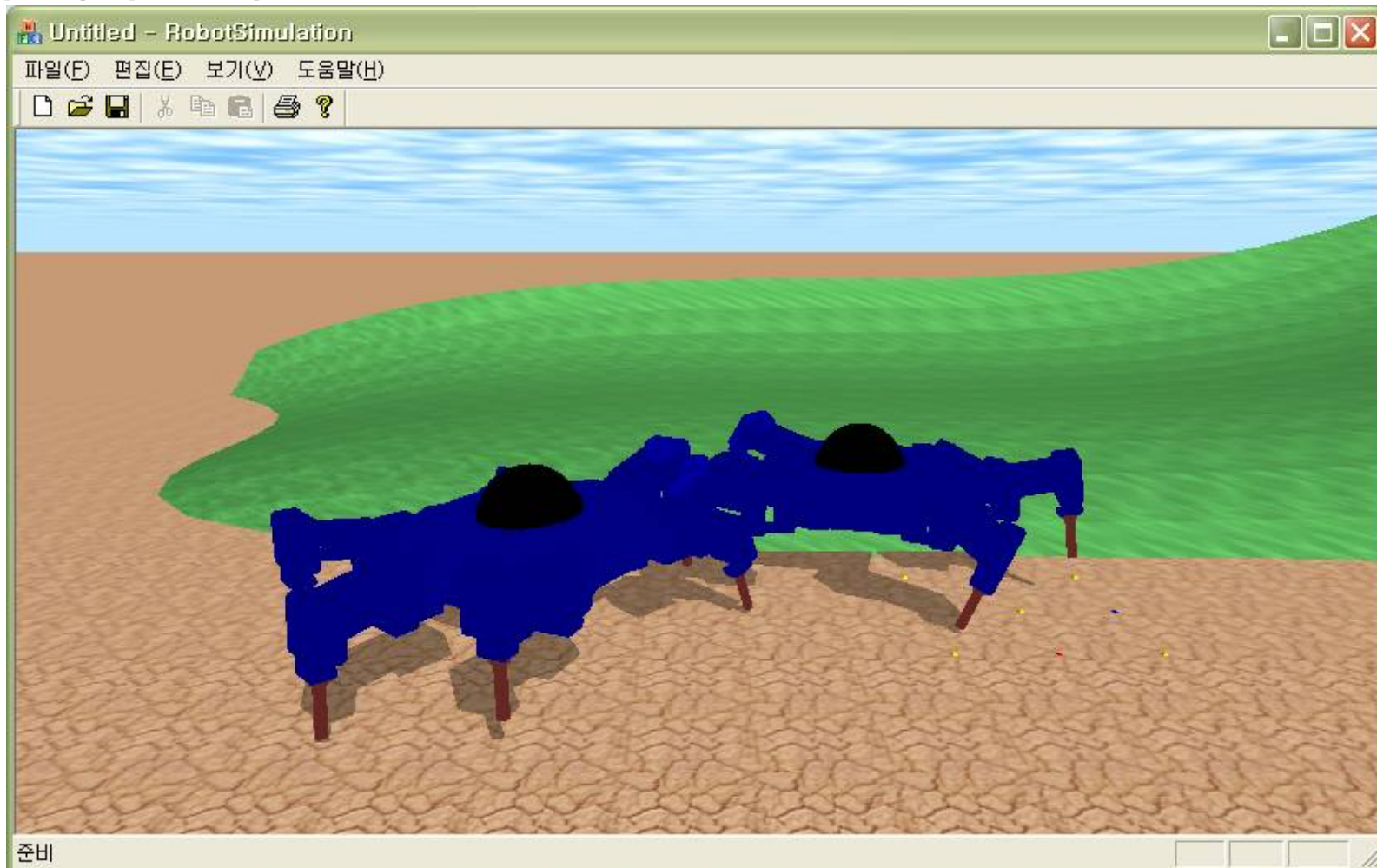
13

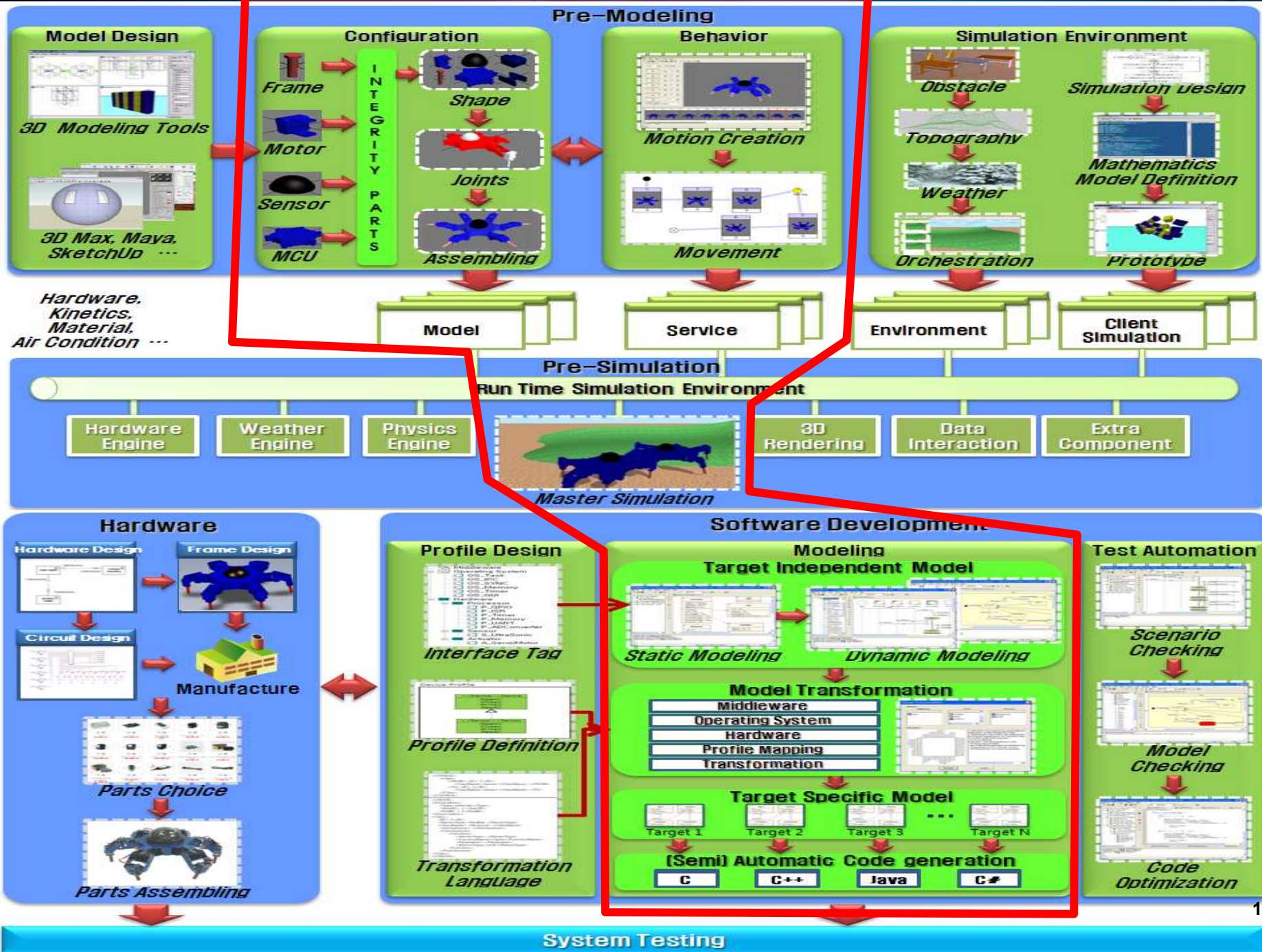
Pre-Modeling & Pre-Simulation



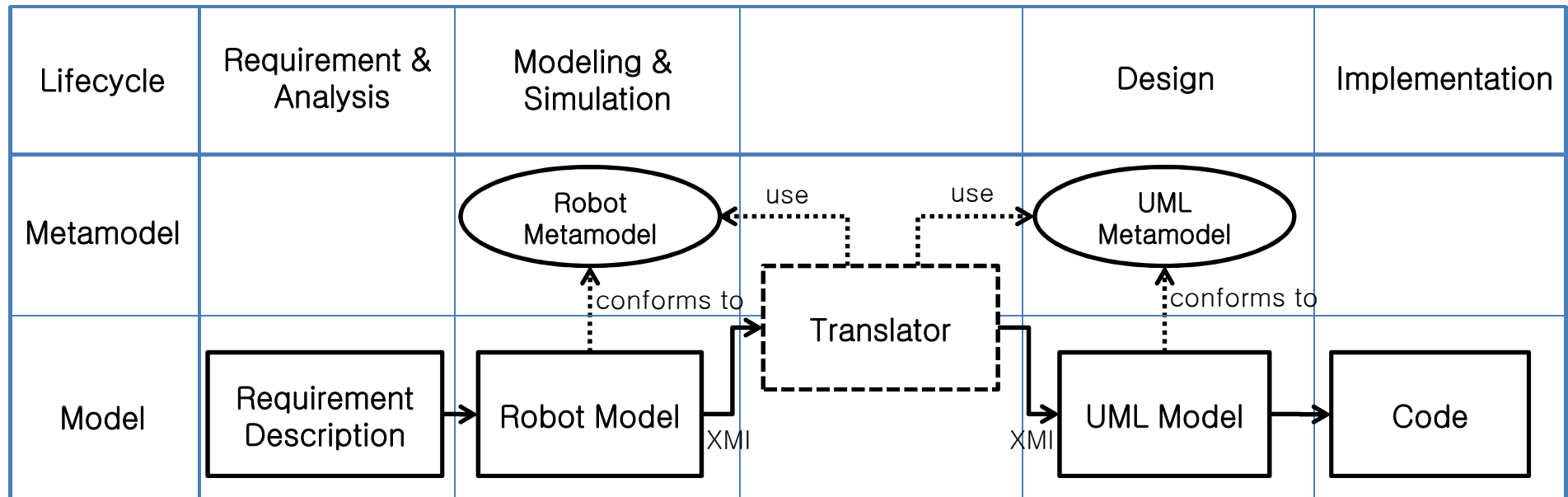
Pre-Simulation

❖ 시뮬레이션 도구

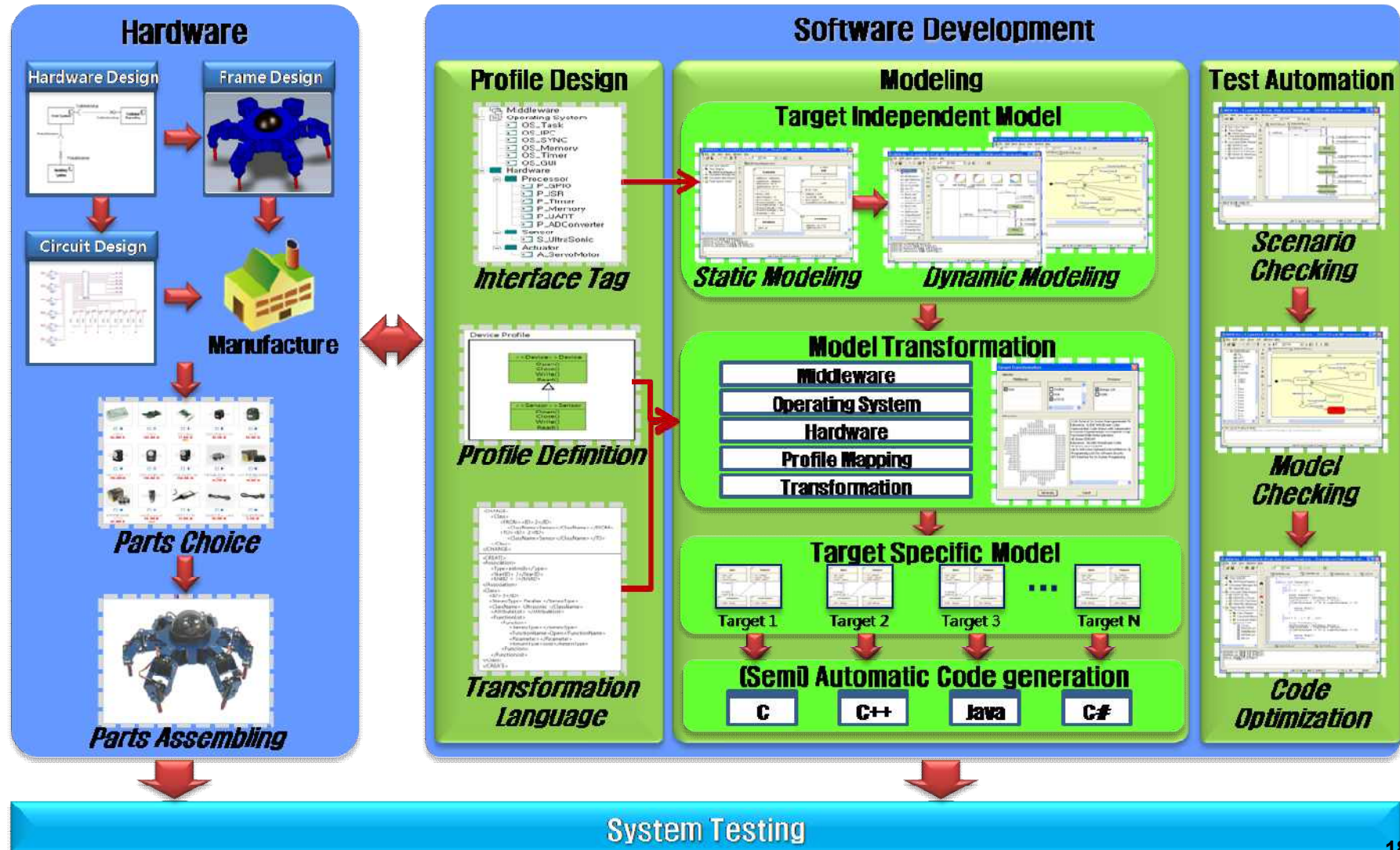




Modeling & Simulation → TIM 모델 변환



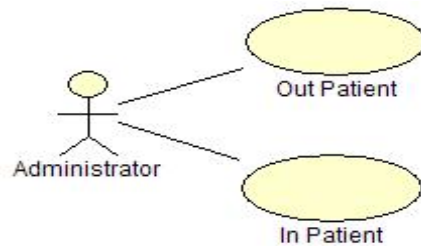
Software Development



모델링 절차

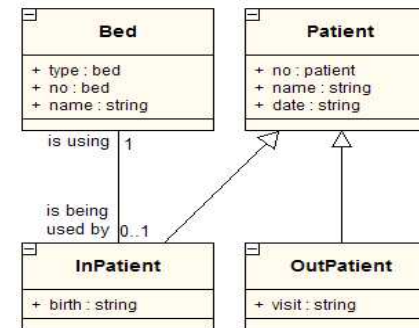
1. 요구사항 명세화

Use Case Diagram

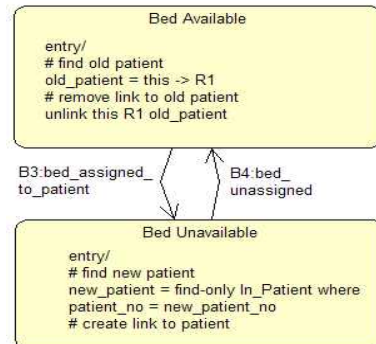


2. 클래스 명세화

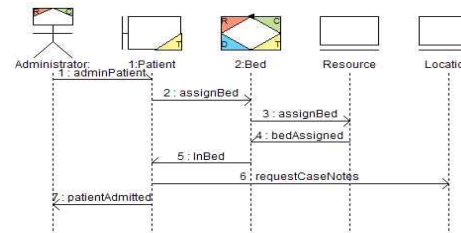
Class Diagram



Concurrent State Diagram



Concurrent Message Diagram

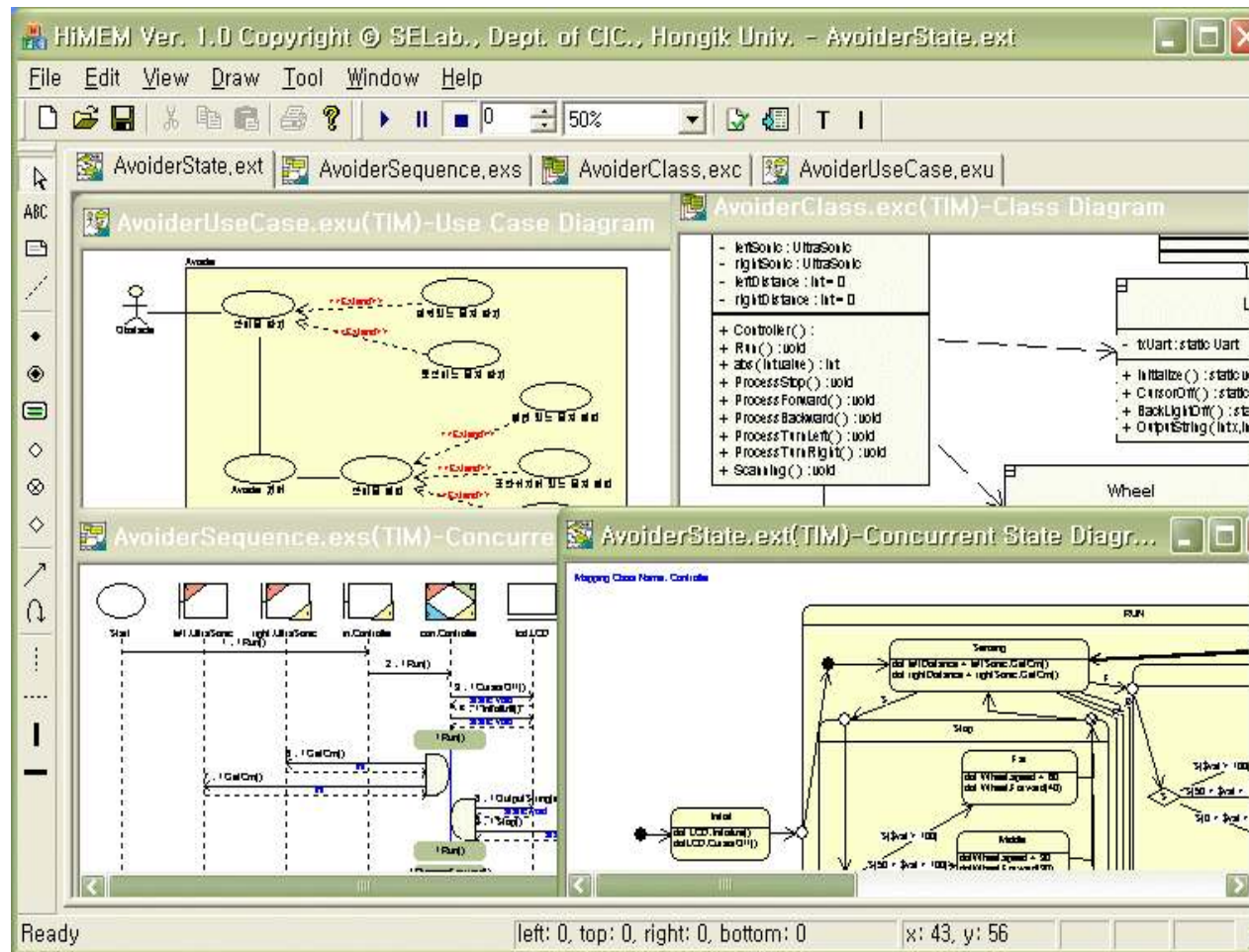


4. 행위 명세화

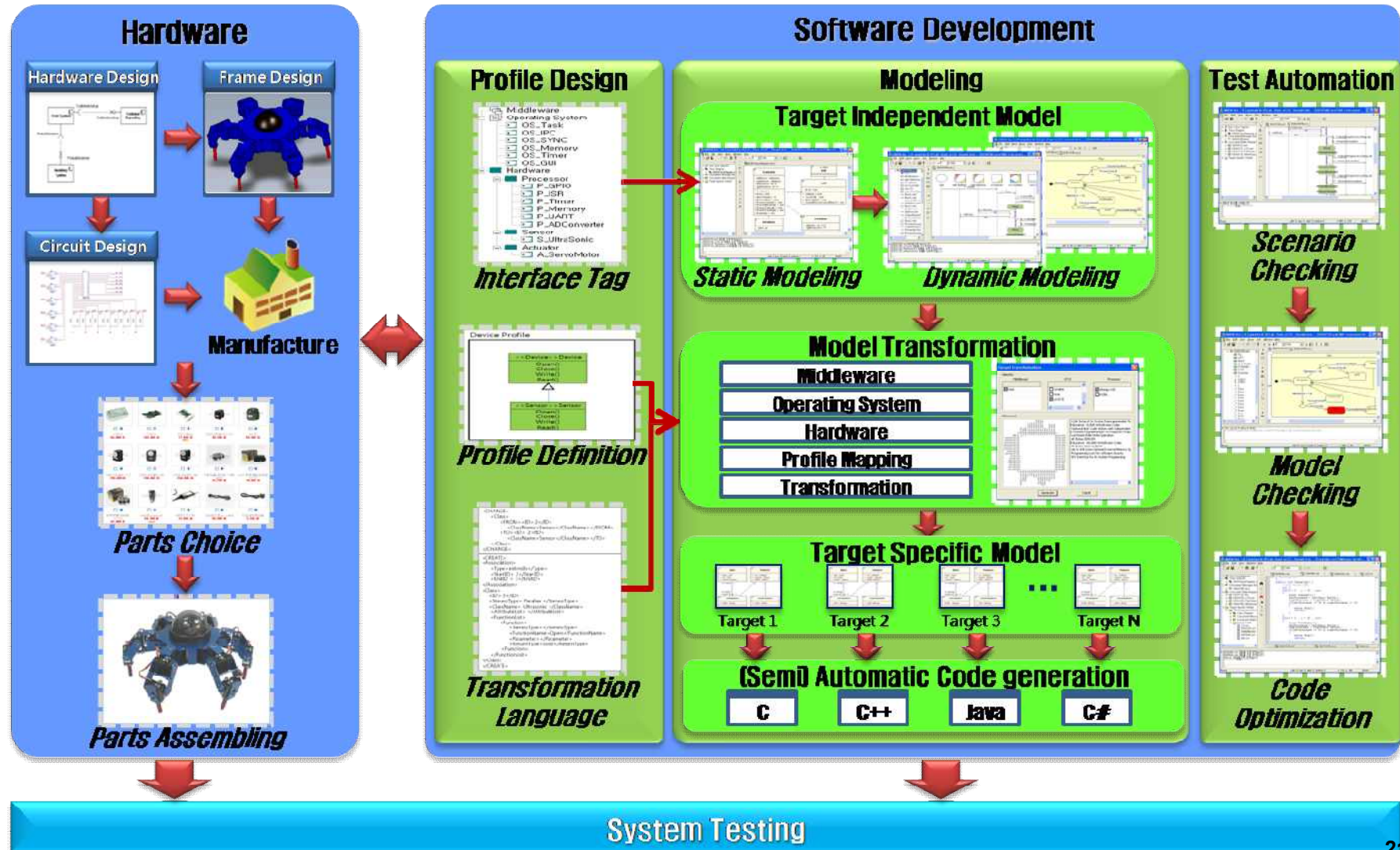
3. 객체 상호작용 명세화

다관절 로봇 소프트웨어 개발 도구

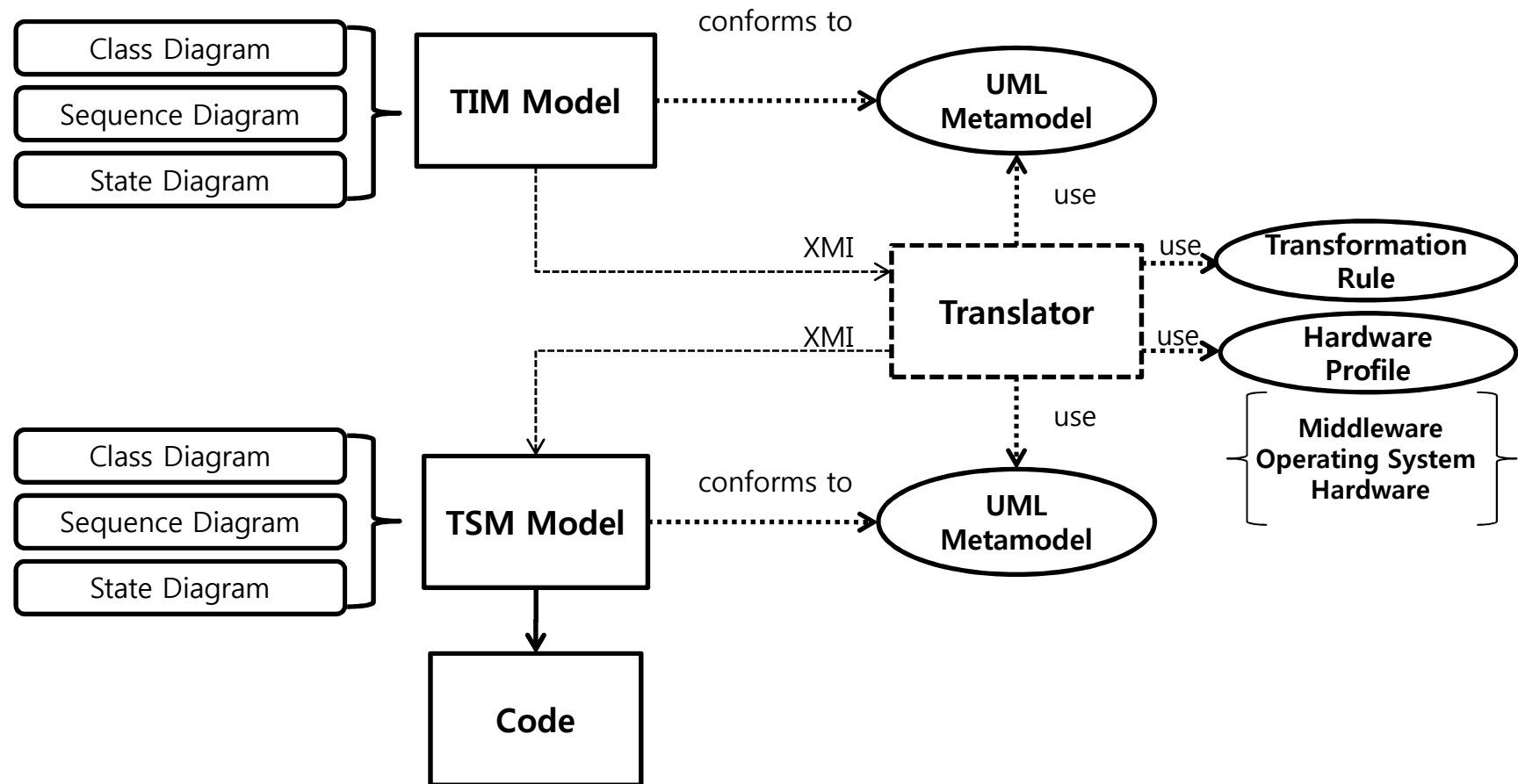
❖ UML Tool



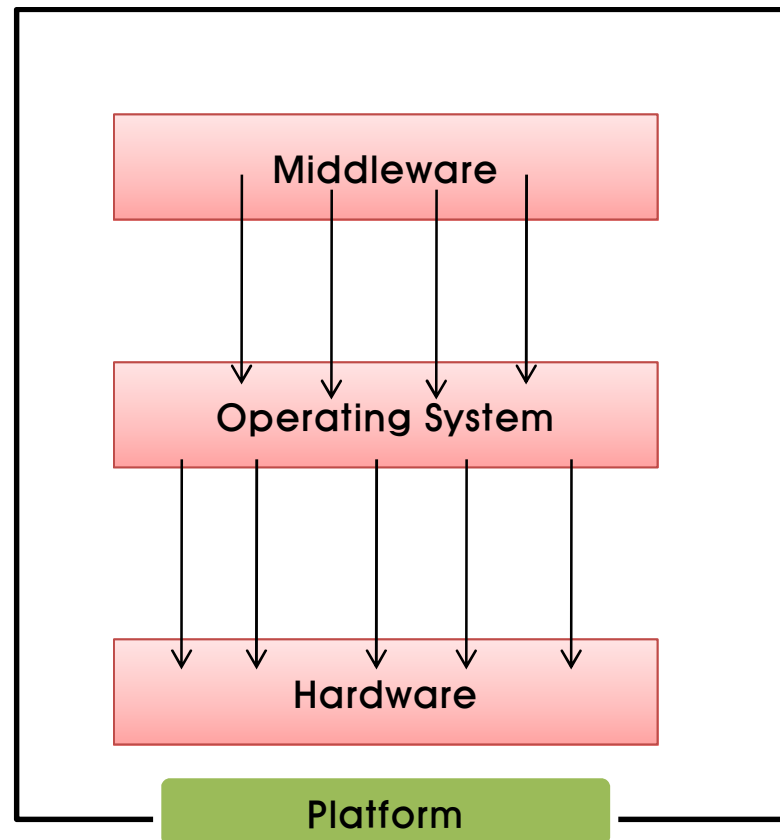
Software Development



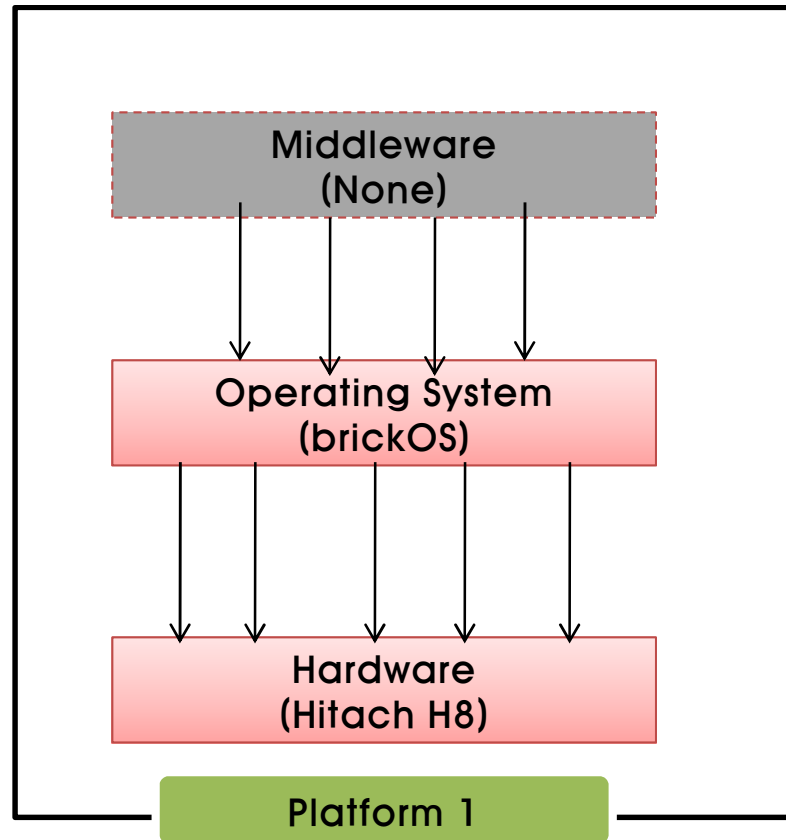
모델 변환 메카니즘



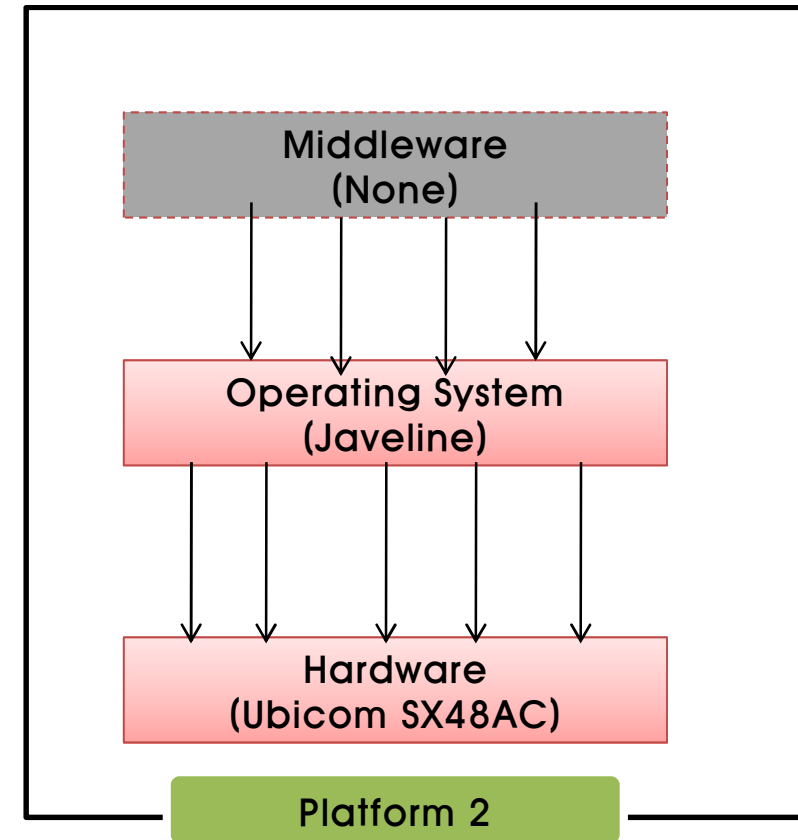
임베디드 시스템 플랫폼



임베디드 시스템 플랫폼의 예

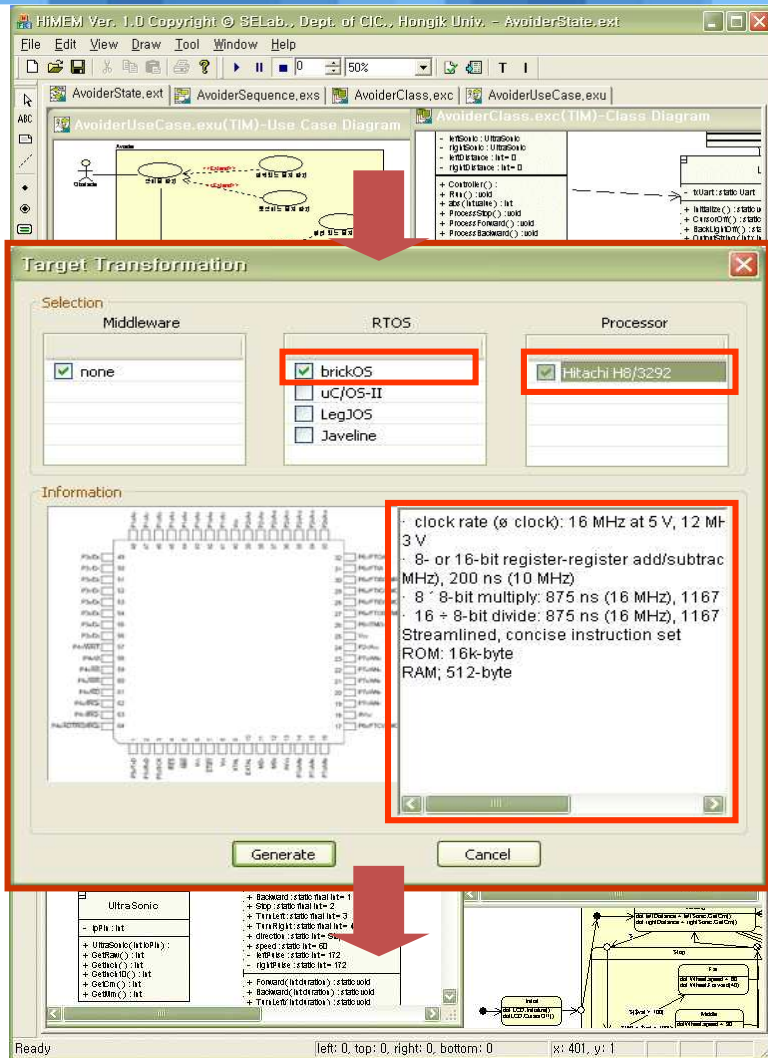


(a) System 1

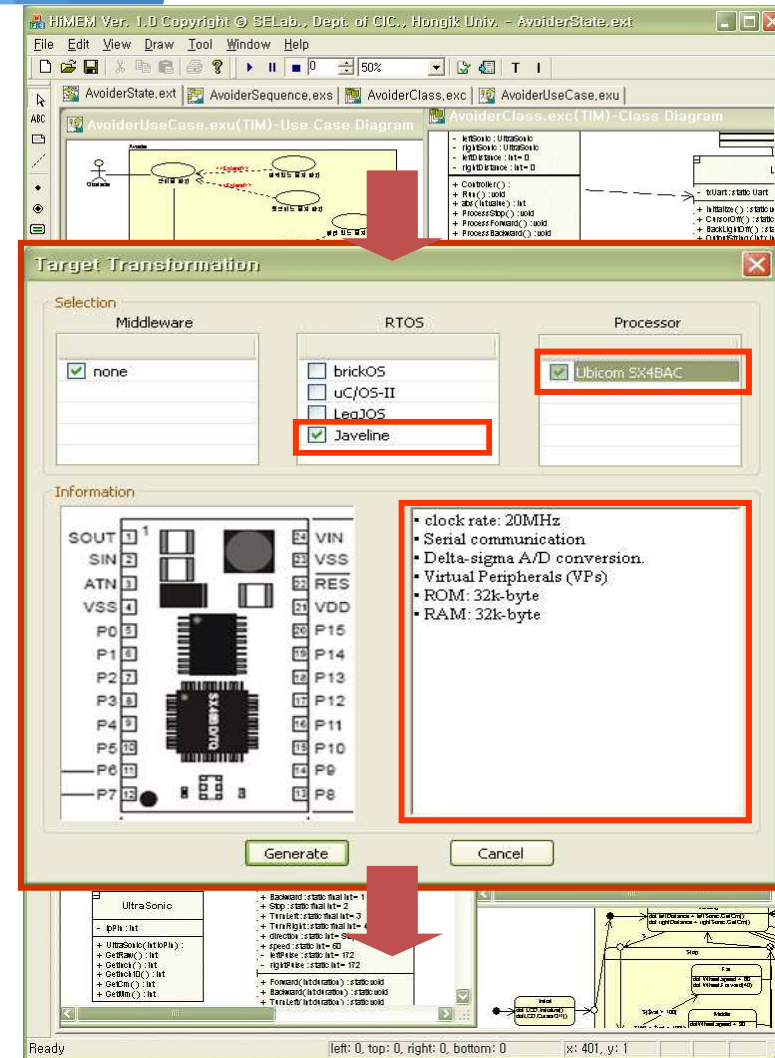


(b) System 2

TIM → TSM 변경 - 모델 변환

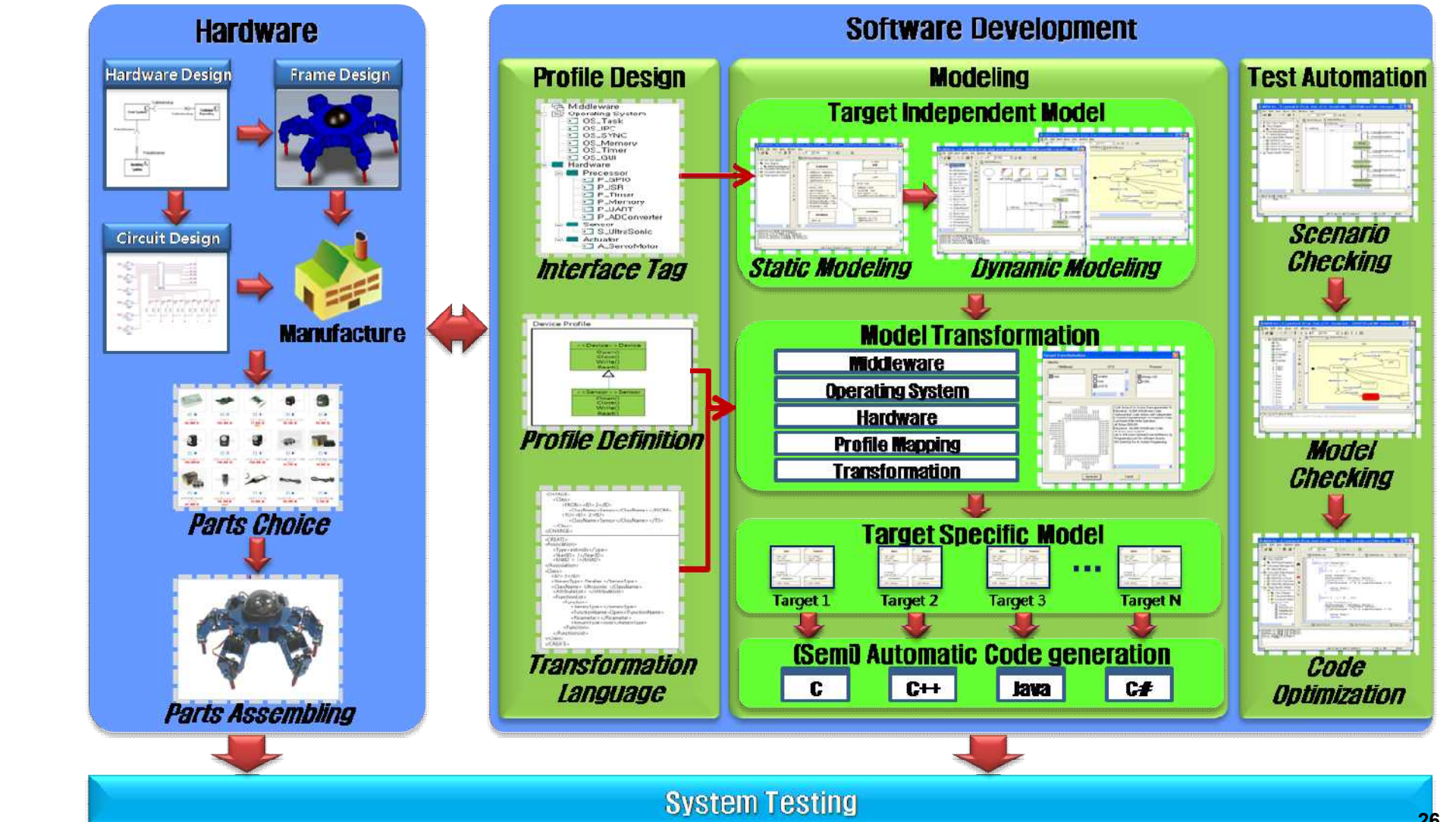


(a) System 1



(b) System 2

Software Development



타겟 의존 코드 생성

Class Name: string	
Package List: List	
Parent List : List	
Interface List : List	
Association List: List	
Association List	Attribute
	SetFunction
Attribute List: List	
Function List: List	
Function List	Head
	Body

코드 메타데이터

Class Name: string

Package List: List

Parent List : List

Interface List : List

Association List: List

Association List Attribute

SetFunction

Attribute List: List

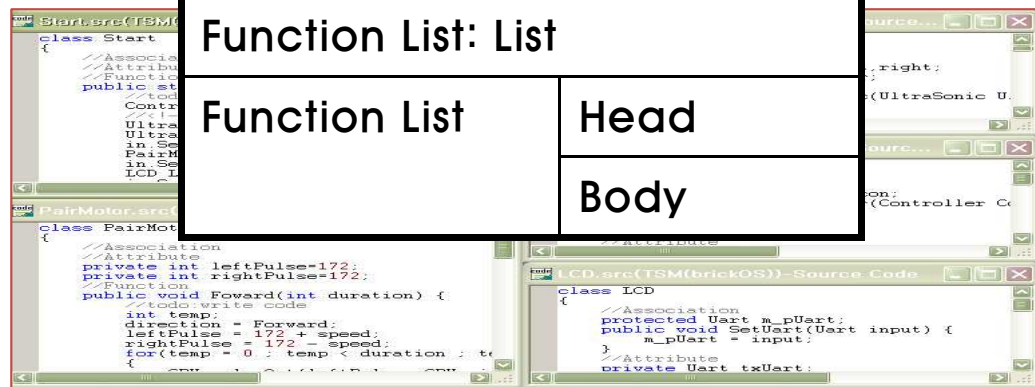
Function List: List

Function List Head

Body

C	
#include [Package List]	
//attribute	
[Attribute List]	
//Function	
[Function List(Head)]	
{	
[Function List(Body)]	
}	

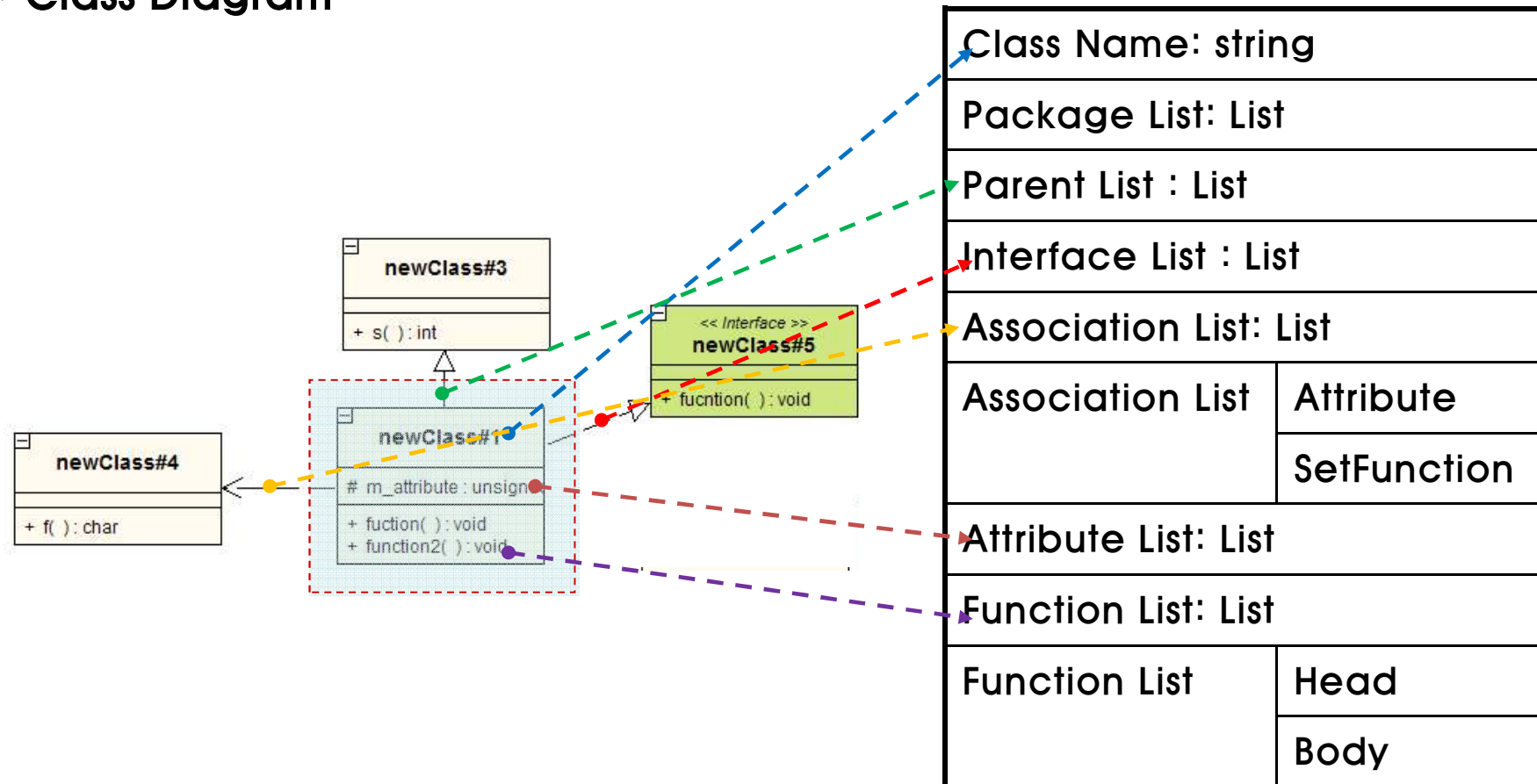
플릿



생성된 코드

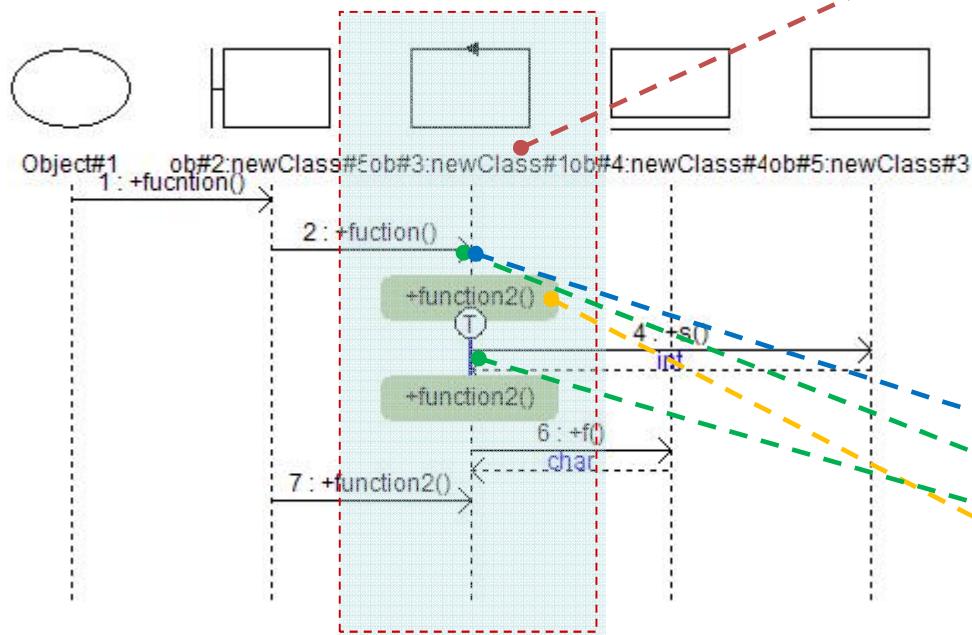
코드 생성(Class Diagram 매핑)

❖ Class Diagram



코드 생성(Concurrent Message Diagram 매핑)

❖ Concurrent Message Diagram

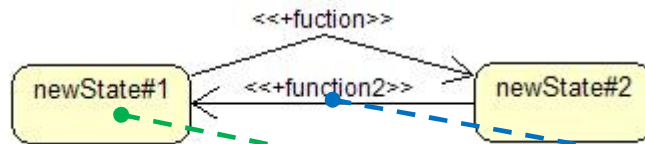


Class Name: string	
Package List: List	
Parent List : List	
Interface List : List	
Association List: List	
Association List	Attribute
	SetFunction
Attribute List: List	
Function List: List	
Function List	Head
	Body

코드 생성(Concurrent State Diagram 매핑)

❖ Concurrent State Diagram

Mapping Class Name: newClass#



Class Name: string	
Package List: List	
Parent List : List	
Interface List : List	
Association List: List	
Association List	Attribute
	SetFunction
Attribute List: List	
Function List: List ▶	
Function List	Head
	Body

코드 생성 (코드 메타데이터와 코드템플릿 매핑)

❖ 코드 메타데이터(Class, Message, State)

Class Name: string	
Package List: List	
Parent List : List	
Interface List : List	
Association List: List	
Association List	Attribute
	SetFunction
Attribute List: List	
Function List: List	
Function List	Head
	Body

Source Code(java)

```
import stamp.core.*;
class newClass#1 extends newClass#3 implements newClass#5
{
    //Association
    protected newClass#4 ob#4;
    public void SetnewClass#4(newClass#4 newClass#41) {
        ob#4 = newClass#41;
    }
    //Attribute
    protected unsigned short m_attribute;
    //Function
    public void fuction() {
        //todo:write code
    }
    public void function2() {
        //todo:write code
        m_attribute="test";
    }
};
```

코드 생성(각각의 코드생성 템플릿)

Java

```
import [Package List]
class [Class Name]
extends [Parent List]
implements [Interface List]
{
    //association
    protected
    [Association List(Attribute)]
    public
    [Association List(SetFunction)]
    //attribute
    [Attribute List]
    //Function
    [Function List(Head)]
    {
        [Function List(Body)]
    }
};
```

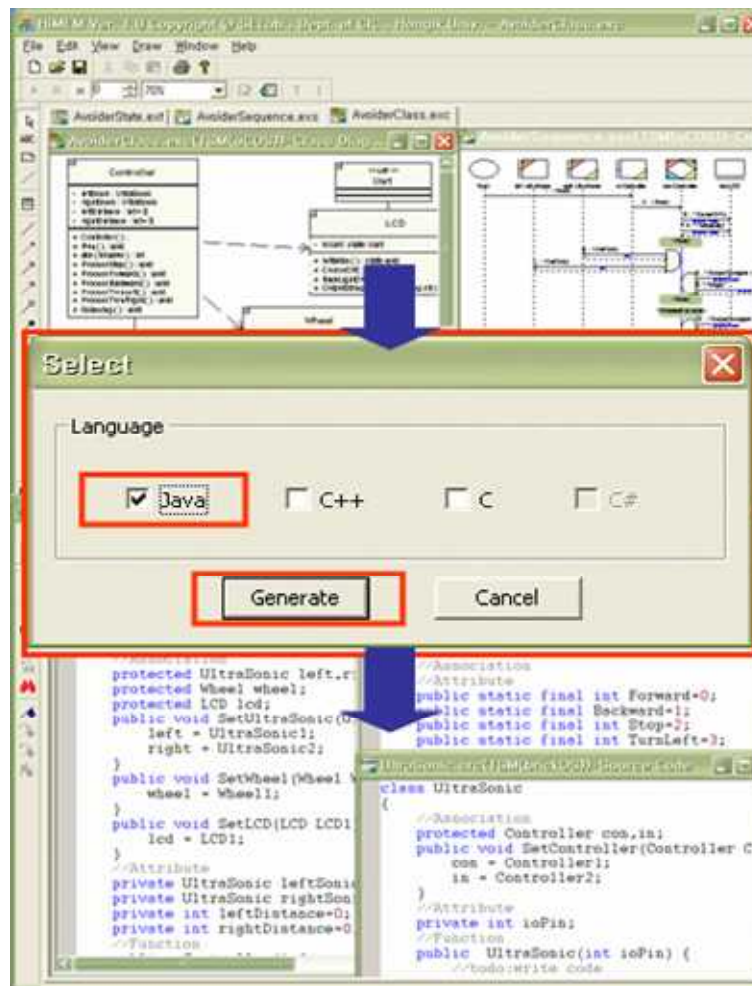
C++

```
#include [Package List]
class [Class Name]
: [Parent List], [Interface List]
{
    //association
    protected :
        [Association List(Attribute)]
    public :
        [Association List(SetFunction)]
    //attribute
    [Attribute List]
    //Function
    [Function List(Head)]
    {
        [Function List(Body)]
    }
};
```

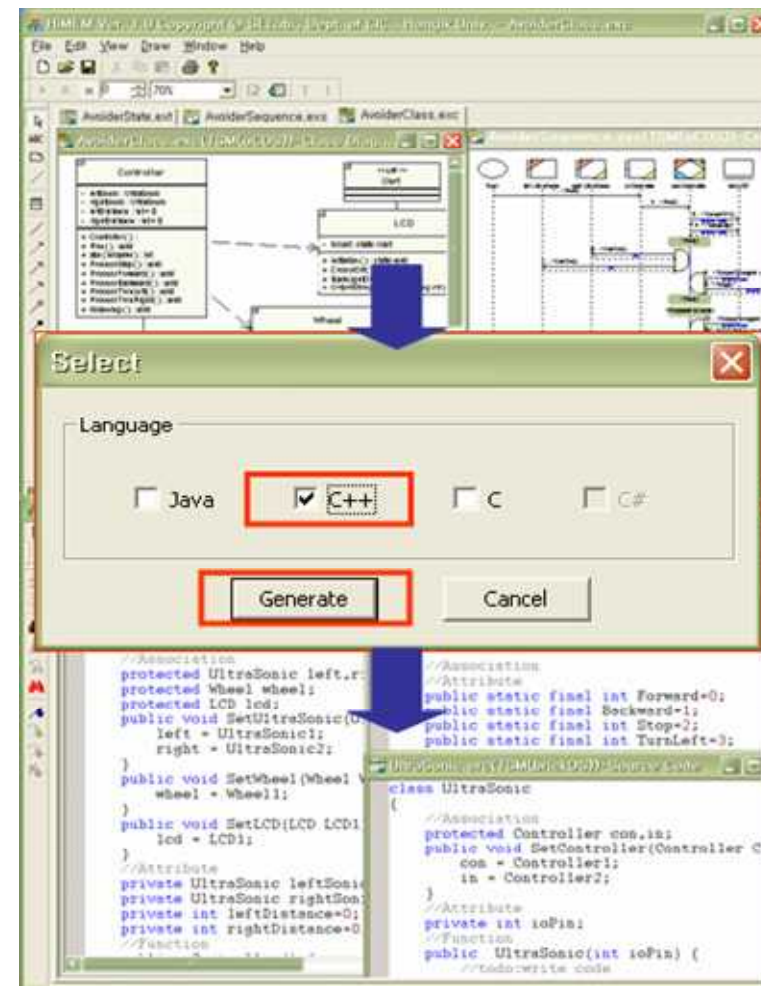
C

```
#include [Package List]
//attribute
[Attribute List]
//Function
[Function List(Head)]
[Function List(Head)]
{
    [Function List(Body)]
}
```

TIM → TSM 변경 - 코드생성



(a) System 1



(b) System 2

도구 비교

		Tau	Rose RT	Rhapsody	HiMEM
Primary Market		Telecom	Real-time	Embedded	Embedded, Real-time
Product Code Generation	Code Generation	70~90%	70~80%	80~85%	80~90%
	Readable Codes	No	Yes	Yes	Yes
	Source level Debugging	Absent	Supported	Supported	Supported
	Rules based Code Generation	Absent	Supported	Supported	Supported
Roundtrip Engineering	Model/Code Associativity	Supported, but some restrictions	Supported, but some restrictions	Supported	Supported
	Dynamic Model Code View	Absent	Absent	Supported	Supported
	Model/Code Synchronization	Absent	Absent	Supported	Supported
Reverse Engineering	Reverse Engineering	Supported	Supported	Supported	Absent
	Forward Generation	Supported	Supported	Supported	Supported
Model Execution	Executable Model	Supported	Supported, but using by VM	Supported	Supported
	Simulation	Supported	Supported	Supported	Supported
	Simulating Diagrams	Sequence, Statechart	Sequence, Statechart	Sequence, Statechart, Activity	CMD, CSD
	Design-level debugging, animation	Absent	Absent	Supported	Supported
	Requirements validation	Supported	Absent	Supported	Supported
	Use Case tracing	Absent	Absent	Supported	Supported
	Reverse Fork/Join	Absent	Absent	Absent	Supported

동영상

향후 적용 사례 1

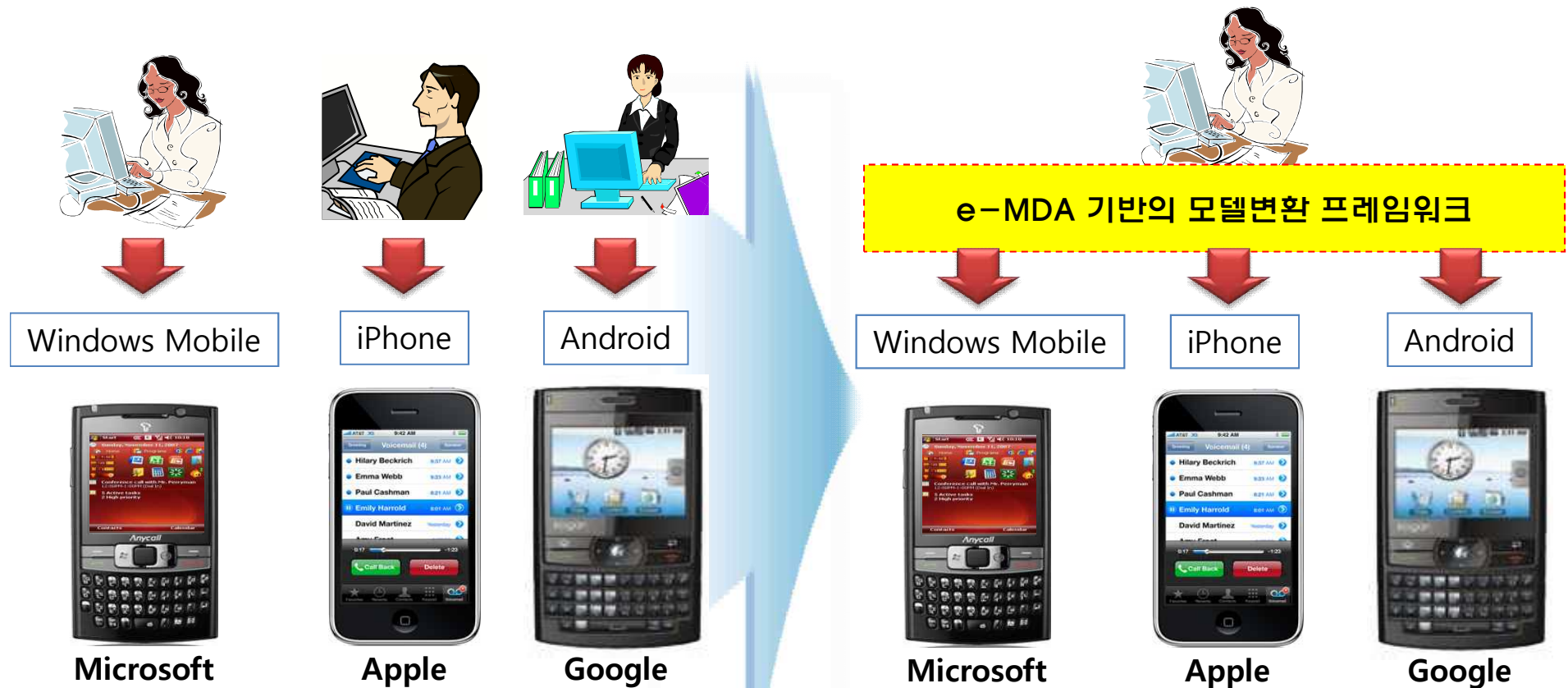
Parallax QuadRover Robot



Hanback HBE-EROBO-TANK



향후 적용 사례 2



지적재산권 현황

❖ M&S 도구

- 국제특허 SPT20090102US ROBOT MOTION DATA GENERATION METHOD AND A GENERATION APPARATUS USING IMAGE DATA
- 국내특허 10-2008-0018493 다관절 로봇 시뮬레이션 제어 프로그램 개발 도구

❖ 임베디드 소프트웨어 개발 도구

- 국제특허 STP20090101US EMBEDDED SYSTEM DESIGN DEVICE AND A METHOD THEREOF USING A TARGETINDEPENDENT MODEL
- 국내특허 10-2008-0120892 타겟독립모델을 이용한 임베디드 시스템 설계 장치 및 그 방법

The End

Thank You !

김 영 철

홍익대학교 컴퓨터정보통신

(041)865-2477, 016-659-7518

소프트웨어공학연구실

bob@hongik.ac.kr <http://selab.hongik.ac.kr>