무인항공기 OFP를 위한 SW V&V 기법

유 준 범
Dependable Software Laboratory
(http://dslab.konkuk.ac.kr)

건국대학교 컴퓨터공학부

Contents

- 무인비행체 개발 프로젝트 개요

- OFP V&V 계획
  - OFP (Operational Flight Program)
  - 정형 모델링 및 검증
  - 역공학
  - SW 테스팅

- Formal Modeling and Verification
- Reverse Engineering
- SW Testing

- 맺음말

무인비행체 개발 프로젝트 개요

소형 무인비행체 개발 프로젝트

- **개발 주체: UAV & SW Fusion Research Center (UAV·SW)**
  - 소속 : 지식경제부 ITRC (IT Research Center) / 건국대학교
  - 책임연구원 : 건국대 김두현 교수
  - 연구기간 : 2008.07 ~ 2011.12

- **연구개발 목표**
  - **무인헬기**를 이용한 **IT 기반 실시간 방재** 및 **IT 서비스** 실현

- **연구개발 성과**
  - 무인비행체(UAVs) 및 내장형제어SW(OFP) : HeliScope , QuadScope
  - 지상제어시스템(GCS) : HeliCommander , HeliNavi , DualEye , HeliVision
  - 시뮬레이터 : HeliHILS , OFP Simulation

KU KONKUK UNIVERSITY

QuadScope GCS

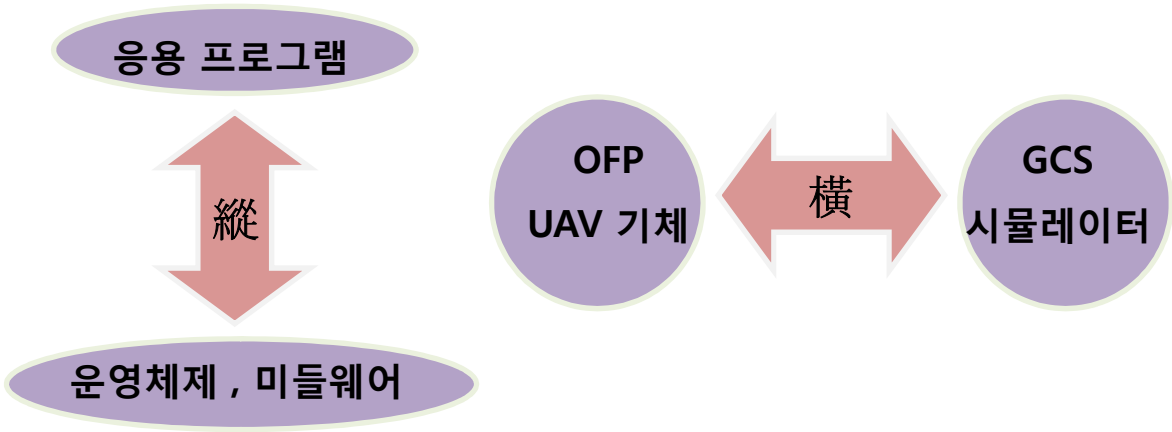
HeliScope

QuadScope

프로젝트 의의

- 항공기 및 지상시스템에서 SW가 큰 역할을 담당
  - 하지만, 부분/개별적으로 개발되면 상호운용이 불가능

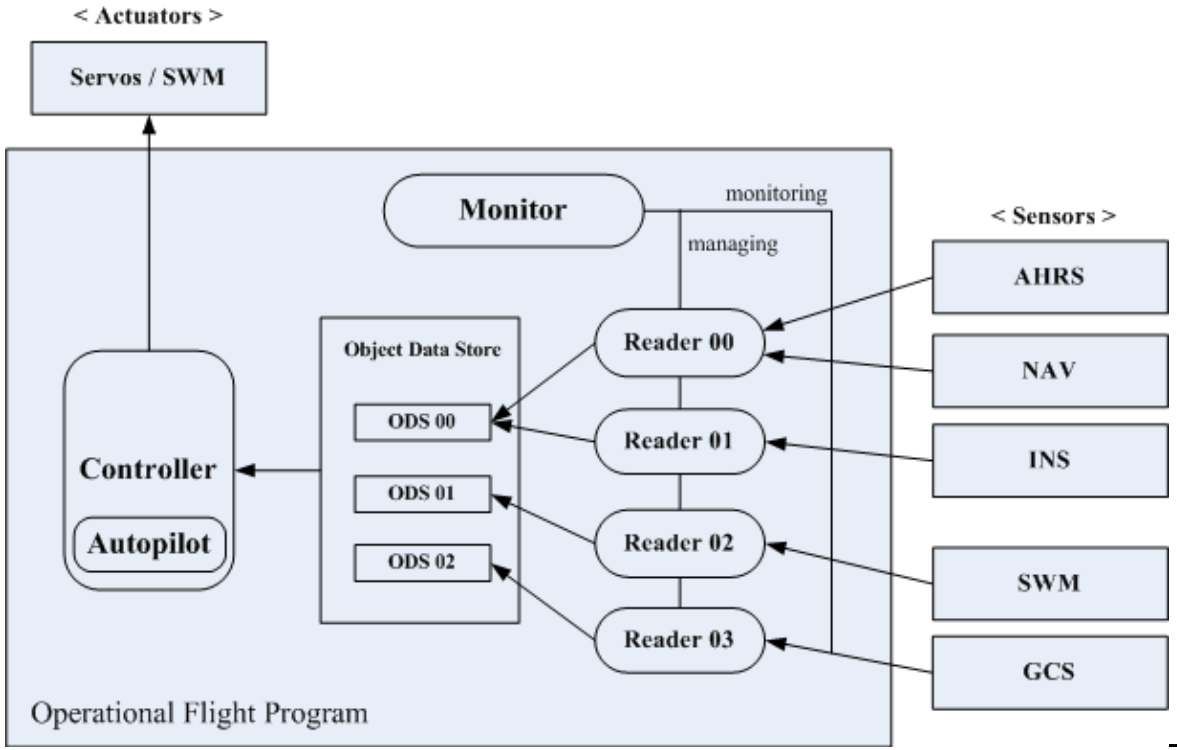
무인항공기/지상제어시스템 全 SW체계 개발



OFP V&V 계획

OFP (Operational Flight Program)

- **무인비행체의 Real-Time Embedded Control Software**
  - 6 프로세스 (1 Controller, 1 Monitor, 4 Reader) , 3 공유 데이터 영역 (ODS)
  - TMO (Time triggered Message) 기반으로 동작

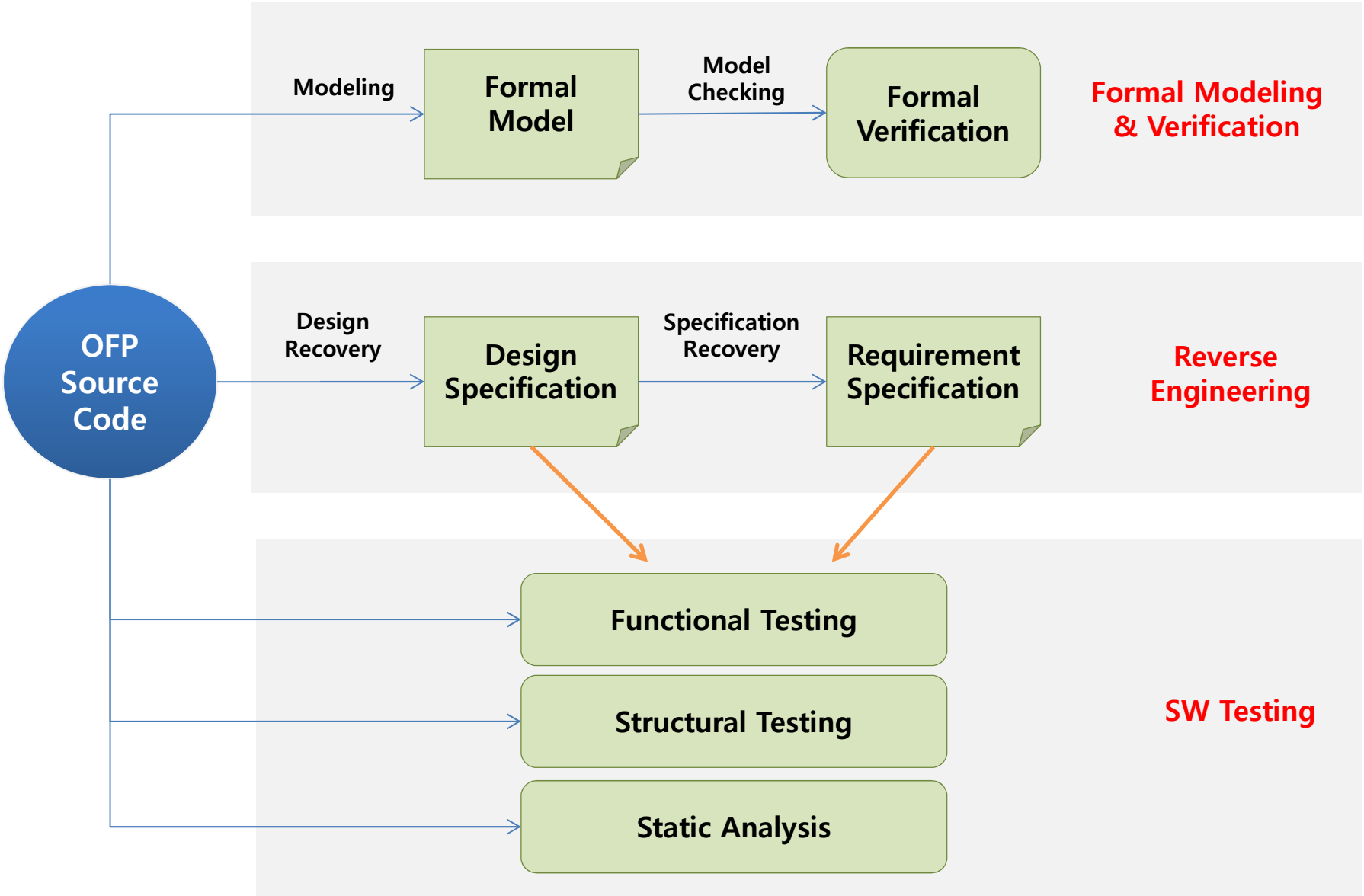


OFP V&V 계획

- **SW 全 개발 Lifecycle에 대한 SW Verification & Validation 수행**
  - 부족한 문서를 보완하기 위해, Reverse Engineering 실시
  - 모델 / 코드 기반으로 구분하여 적용
  - DO-178B 준용

- **DO-178B 등 인증표준 준용**
  - Level A
  - 기능시험(Functional Testing)과 구조시험(Structural Testing)을 병행 수행
  - 코드분석(Static Analysis) 기법 적용

- **정형기법 적용**
  - OFP 정형 모델 개발
  - 정형검증 수행

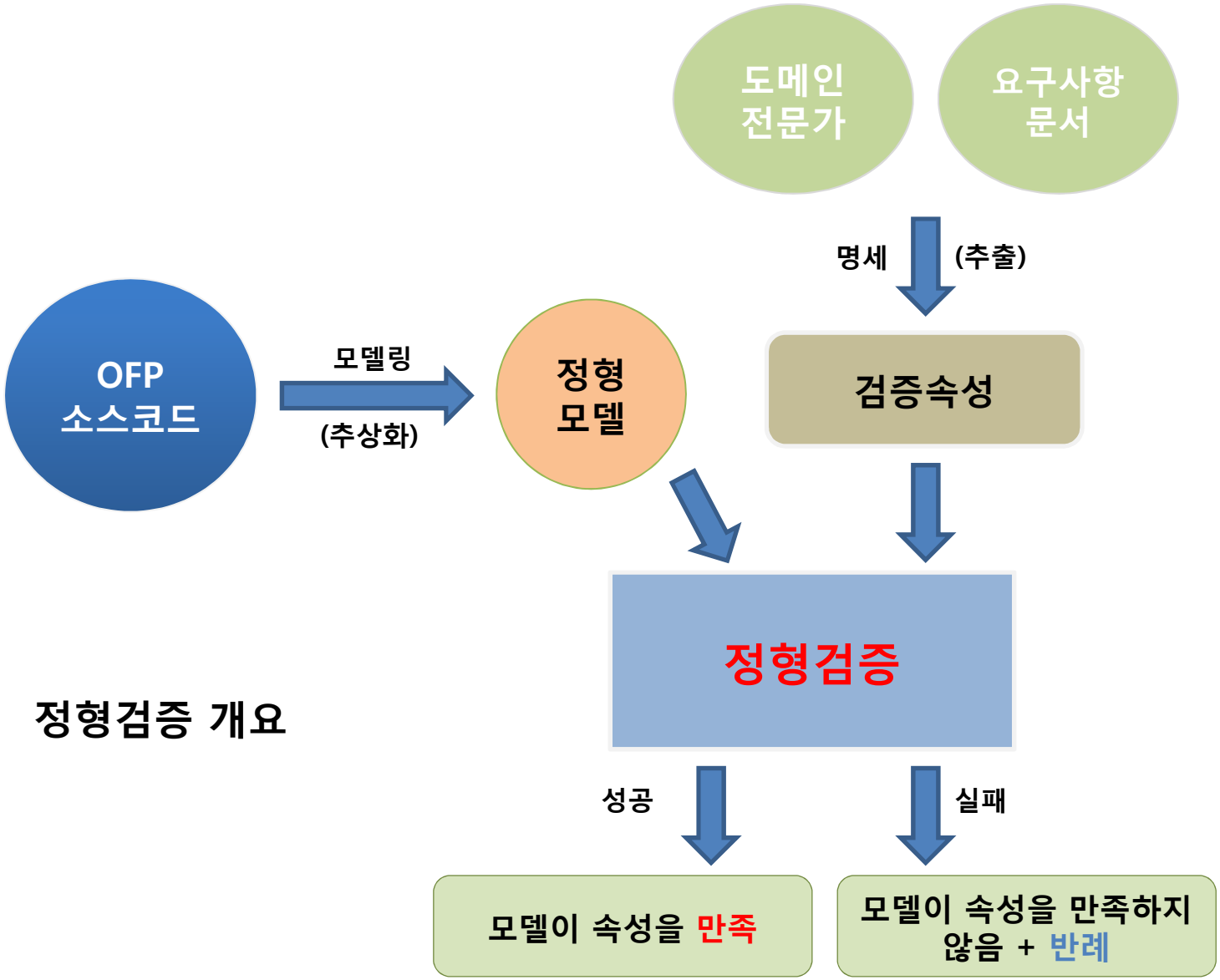

OFP Source Code
Modeling
Formal Model
Model Checking
Formal Verification
Formal Modeling & Verification
Design Recovery
Design Specification
Specification Recovery
Requirement Specification
Reverse Engineering
Functional Testing
Structural Testing
Static Analysis
SW Testing

1. 정형 모델링 및 검증

- **정형 모델링 및 검증**
  - Formal modeling (specification) and verification
  - 정형 모델링: 다양한 FSM (Finite State Machine) 으로 시스템 행위를 정의
  - 정형 검증: 정형모델의 특정 요구사항 만족 여부를 수학적으로 검사

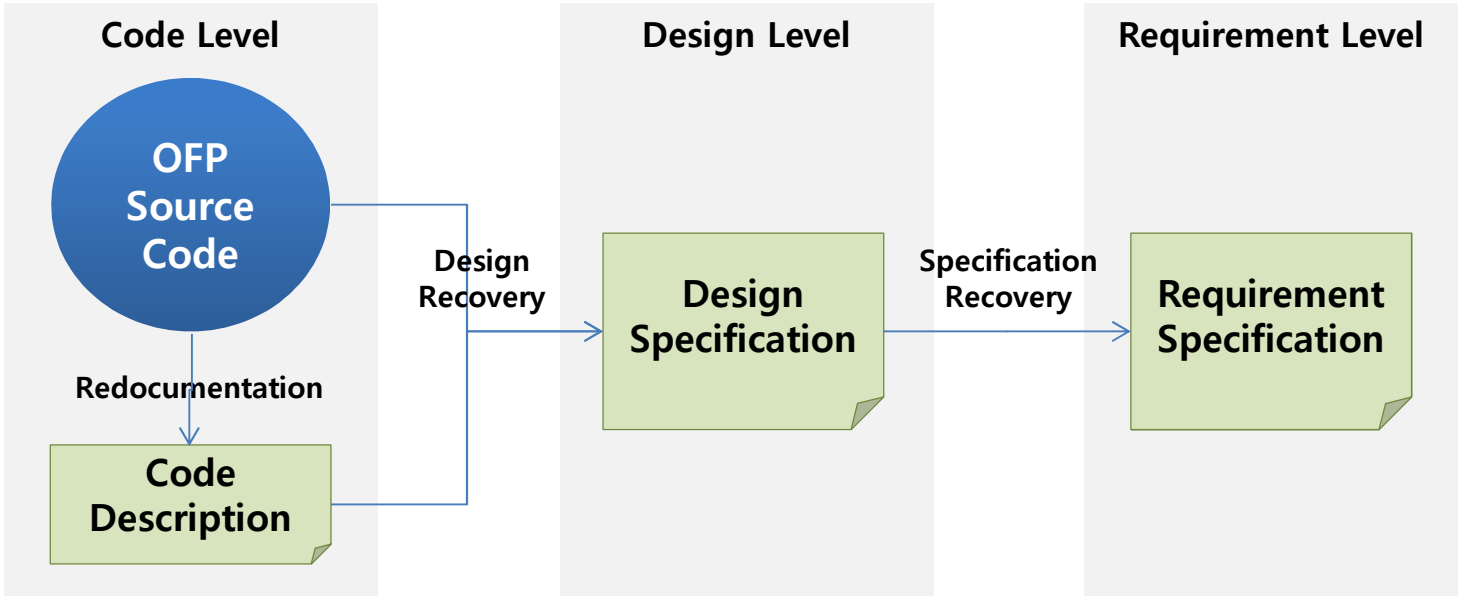
- **OFP의 특성을 고려하여 정형 모델링 및 검증 기법을 선정**
  - 프로세스 間 통신
  - 실시간성
  - 미션 수행
  - 내장형 소프트웨어
  - 코드 중심으로 개발 진행

정형검증 개요


도메인
전문가
요구사항
문서
명세
(추출)
OFP
소스코드
모델링
(추상화)
정형
모델
검증속성
정형검증
성공
실패
모델이 속성을 만족
모델이 속성을 만족하지
않음 + 반례

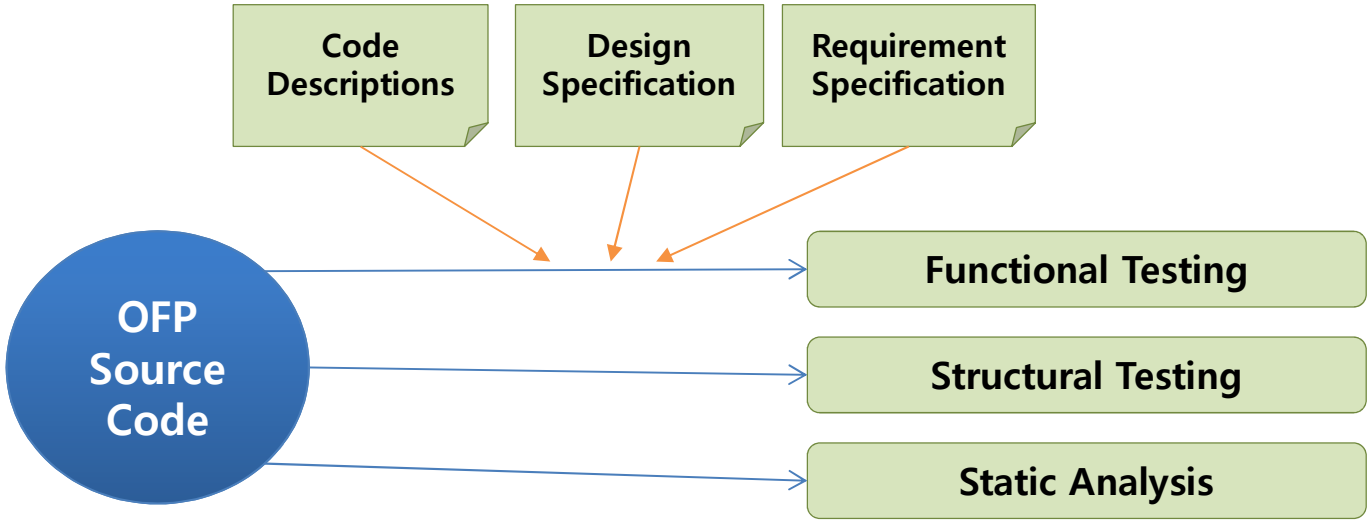
2. 역공학

- **역공학 (Reverse Engineering)을 통한 문서 복구 및 표준화**
  - OFP 개발 중 누락된 개발문서 복구
  - 문서의 표준화 수행
  - V&V 활동의 원천으로 사용
  - 많은 부분을 수작업으로 진행



3. SW 테스팅

- **DO-178B 에 준용하여 SW 테스팅 및 검증기법 적용**
  - 기능 테스트 (Functional Test) : 역공학 결과를 이용
  - 구조 테스트 (Structural Test) : 커버리지 적용, 자동화도구 사용
  - 정적 코드 분석 (Static Analysis) : Checklist 적용



FORMAL MODELING AND VERIFICATION
(정형 모델링 및 검증)

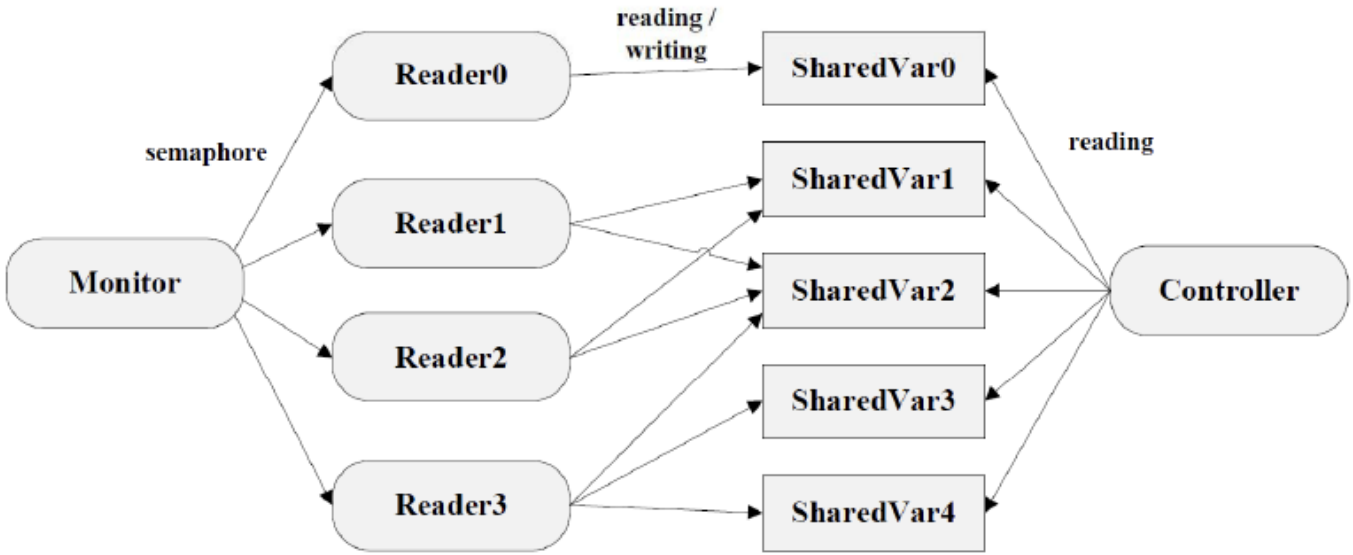
정형검증 도구

- **모델 체커 (Model Checker)**
  - FSM으로 명세된 모델에 정형검증(모델체킹) 알고리즘을 적용하는 도구
  - 다양한 장단점과 특성이 있는 30여 종의 도구들 사용 가능
  - 각 도구 별, 특정 입력 프로그램 언어 사용

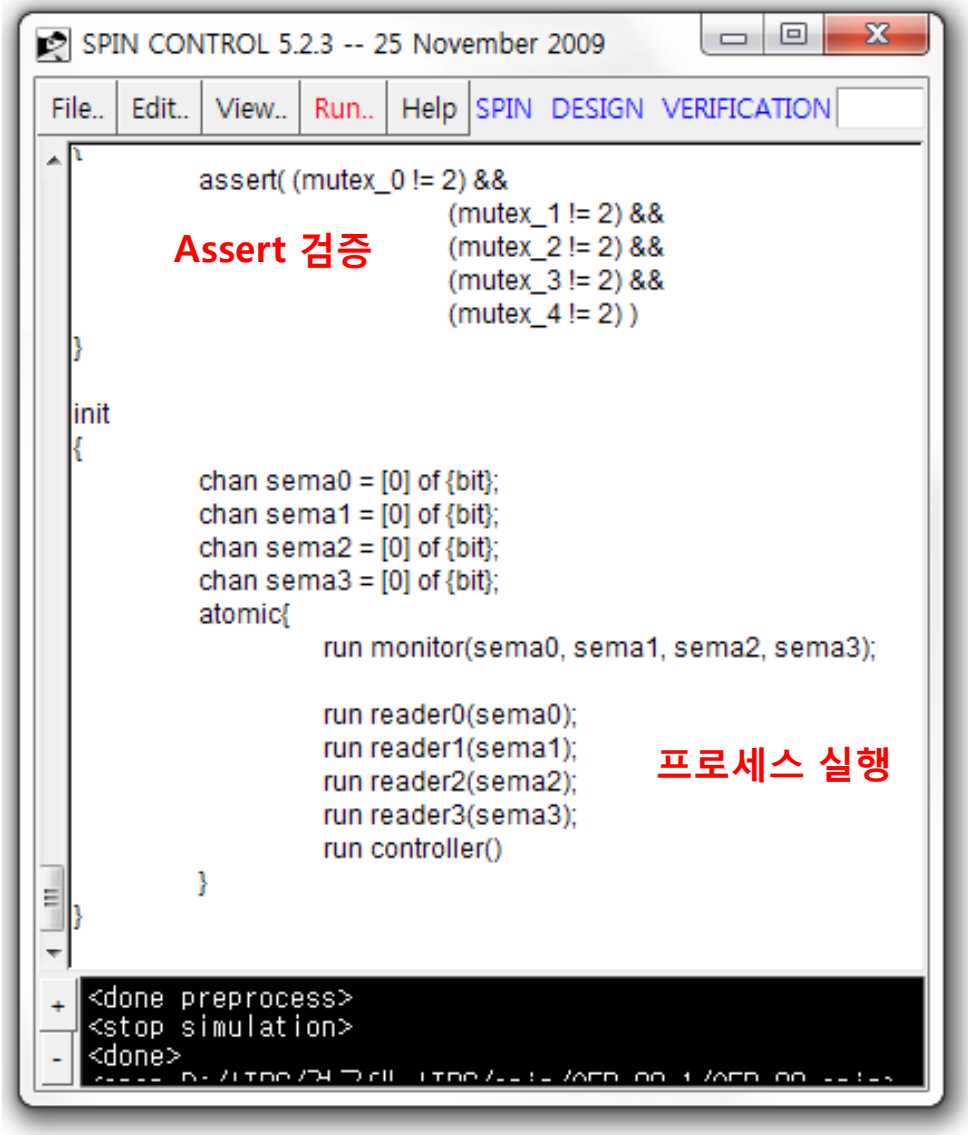
모델체커	특징 (장단점)	입력 프로그램
SMV	컨트롤러 기반의 Reactive system 명세 및 검증	SMV Input Program
SPIN	분산 SW 및 통신 프로토콜 명세 및 검증	PROMELA
VIS	하드웨어 수준의 Synthesis, 시뮬레이션 및 검증	Verilog
CBMC	C 프로그램을 직접 검증 (기능 제한적)	C
UPPAAL	시간을 정확하게 표현해야 하는 시스템의 검증	Timed Automata
HyTech	Hybrid System의 명세 및 검증	Linear Hybrid Automata

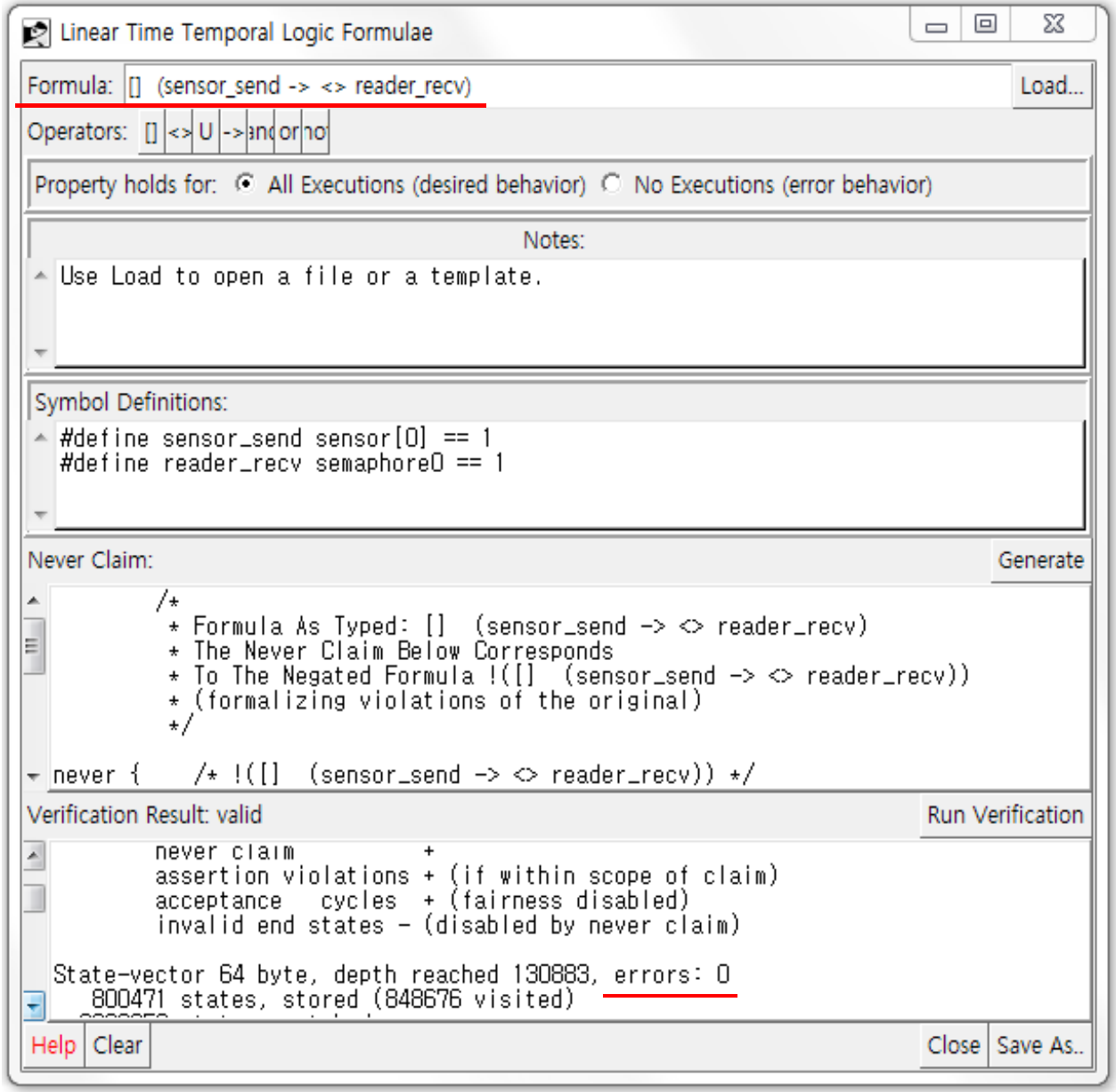
SPIN 정형검증

- **검증내용**
  - "Monitor 프로세스의 세마포어가 정상적으로 동작해야 한다."
    - 검증속성 (LTL Property) : [ ] (sensor_send -> <> read_recv)
  - "Controller와 4개의 Readers가 공유영역을 안전하게 접근해야 한다."
    - SPIN Assert 기능 사용

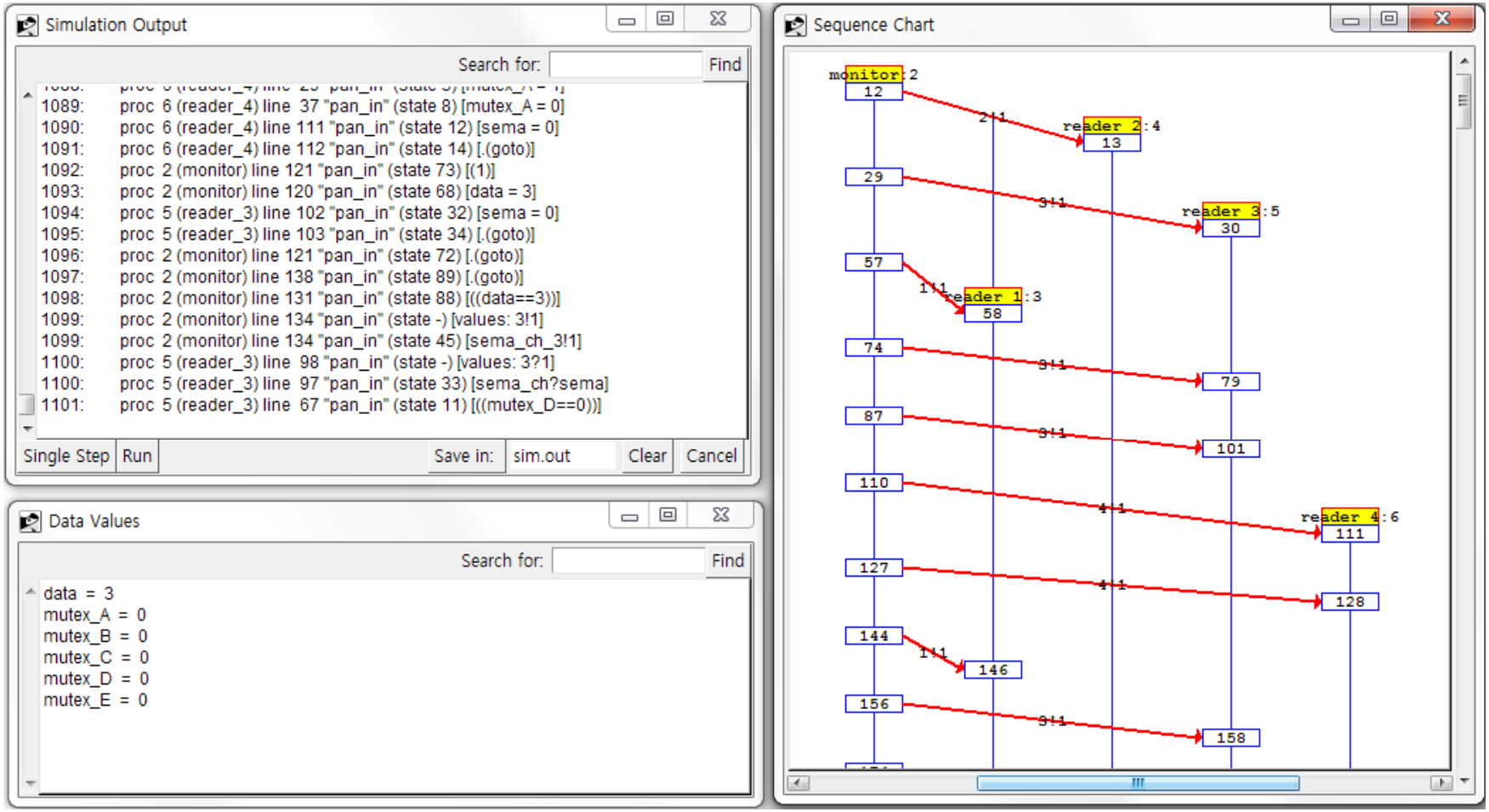


PROMELA 프로그램





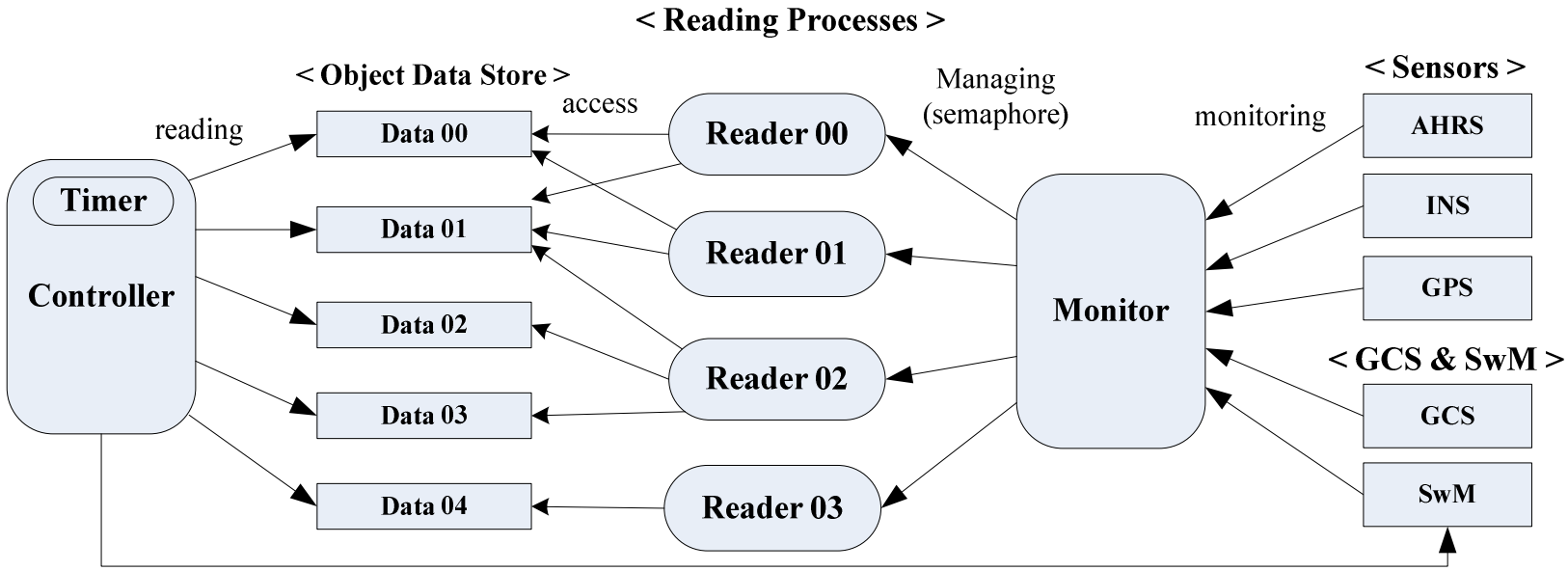
SPIN 실행 결과
(LTL property 검증)

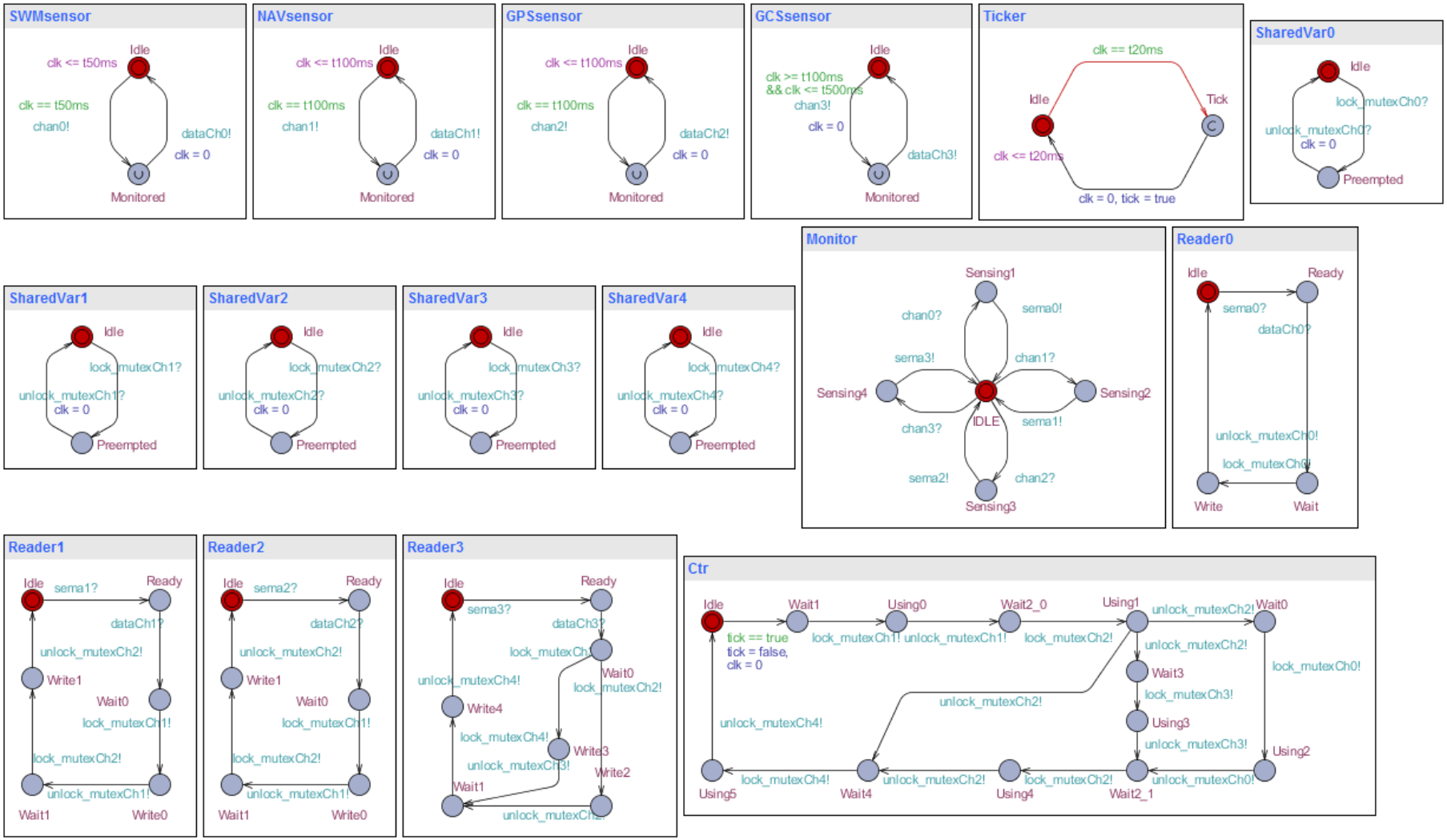


SPIN 시뮬레이션 화면

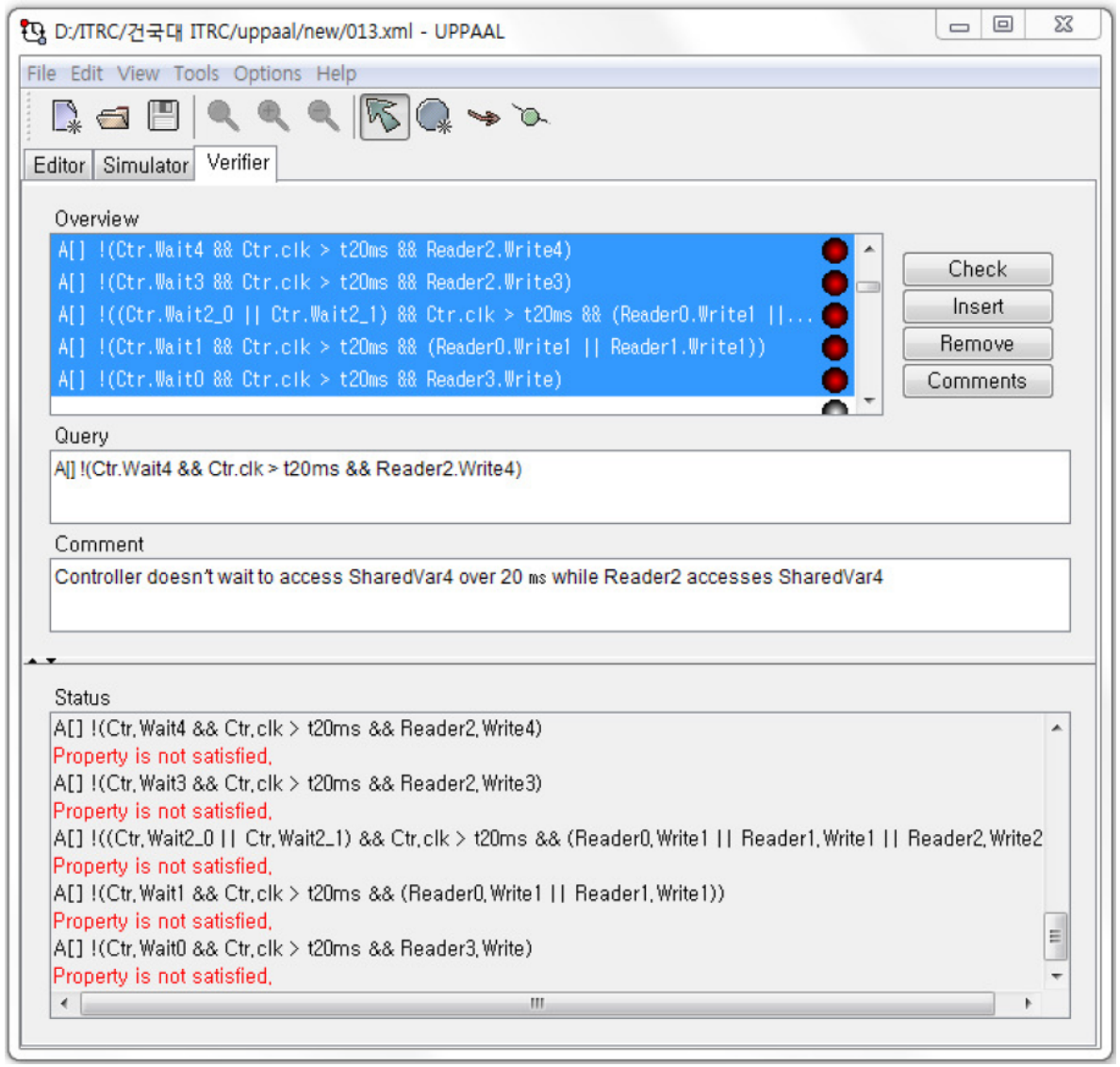
UPPAAL 정형검증

- **검증내용**
  - "각 Reader 프로세스들은 손실 없이 데이터를 읽을 수 있어야 한다."
  - "Controller는 timing bound 내에 데이터를 읽을 수 있어야 한다.
    - 검증속성(CTL Property):
      A□¬ (Ctr.Wait1 && Ctr.clk > t20ms && (Reader0.Write1 || Reader1.Write1))

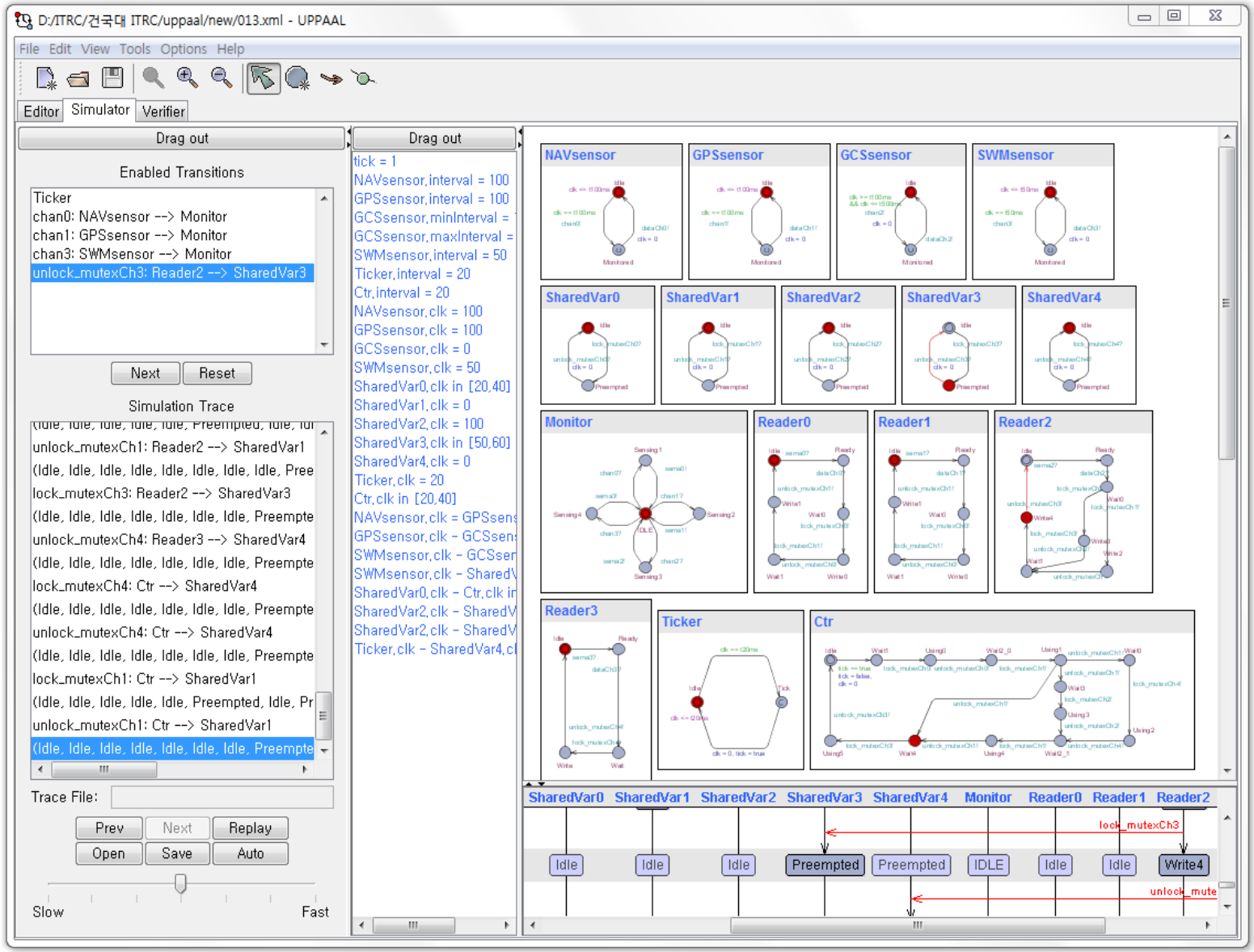




UPPAAL 모델 (Timed Automata)


D:/ITRC/건국대 ITRC/uppaal/new/013.xml - UPPAAL
File Edit View Tools Options Help
Editor Simulator Verifier
Overview
A[] !(Ctr.Wait4 && Ctr.clk > t20ms && Reader2.Write4)
A[] !(Ctr.Wait3 && Ctr.clk > t20ms && Reader2.Write3)
A[] !((Ctr.Wait2_0 || Ctr.Wait2_1) && Ctr.clk > t20ms && (Reader0.Write1 ||...
A[] !(Ctr.Wait1 && Ctr.clk > t20ms && (Reader0.Write1 || Reader1.Write1))
A[] !(Ctr.Wait0 && Ctr.clk > t20ms && Reader3.Write)
Check
Insert
Remove
Comments
Query
A[] !(Ctr.Wait4 && Ctr.clk > t20ms && Reader2.Write4)
Comment
Controller doesn't wait to access SharedVar4 over 20 ms while Reader2 accesses SharedVar4
Status
A[] !(Ctr.Wait4 && Ctr.clk > t20ms && Reader2.Write4)
Property is not satisfied.
A[] !(Ctr.Wait3 && Ctr.clk > t20ms && Reader2.Write3)
Property is not satisfied.
A[] !((Ctr.Wait2_0 || Ctr.Wait2_1) && Ctr.clk > t20ms && (Reader0.Write1 || Reader1.Write1 || Reader2.Write2
Property is not satisfied.
A[] !(Ctr.Wait1 && Ctr.clk > t20ms && (Reader0.Write1 || Reader1.Write1))
Property is not satisfied.
A[] !(Ctr.Wait0 && Ctr.clk > t20ms && Reader3.Write)
Property is not satisfied.

UPPAAL 실행 결과



UPPAAL 시뮬레이션 화면

REVERSE ENGINEERING
(역공학)

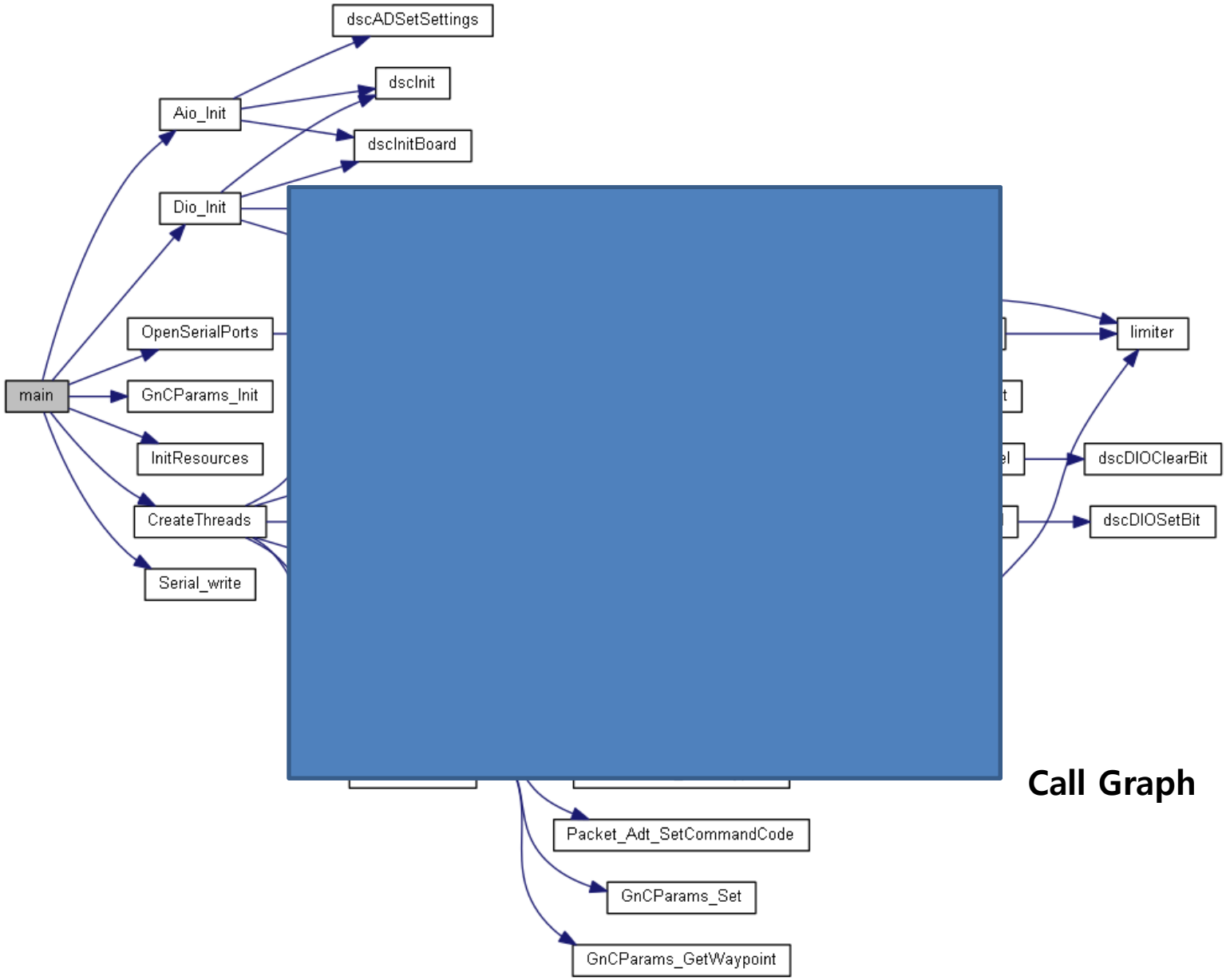
코드 수준의 역공학

- **코드 수준의 문서: Code Description**
  - Data Description
  - Data Structure
  - Call Graph

- **Redocumentation**
  - Source code와 동일한 의미를 지니는 Code description을 생성

Variable	Description	Type or Format
fdDbg	Serial port	int
fdNav	Serial port	int
semAdtReader	Semaphore for data reading from sensor	struct/sem_t
semNavReader	Semaphore for data reading from sensor	struct/sem_t

Data Description


dscADSetSettings
dscInit
Aio_Init
dscInitBoard
Dio_Init
OpenSerialPorts
limiter
main
GnCParams_Init
InitResources
dscDIOClearBit
CreateThreads
dscDIOSetBit
Serial_write
Packet_Adt_SetCommandCode
GnCParams_Set
GnCParams_GetWaypoint

Call Graph

디자인 수준의 역공학

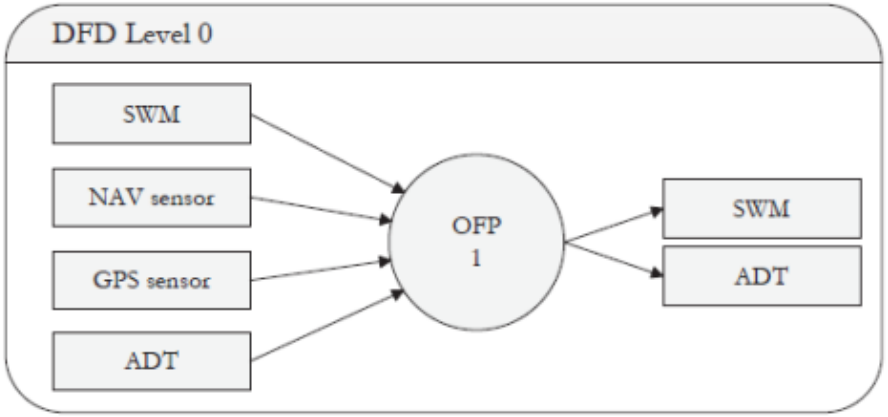
- **디자인 수준의 문서**
  - Design Specification
    - Interface Definitions
    - Data Flow Diagrams (DFD)
    - Control Flow Diagrams (CFD)

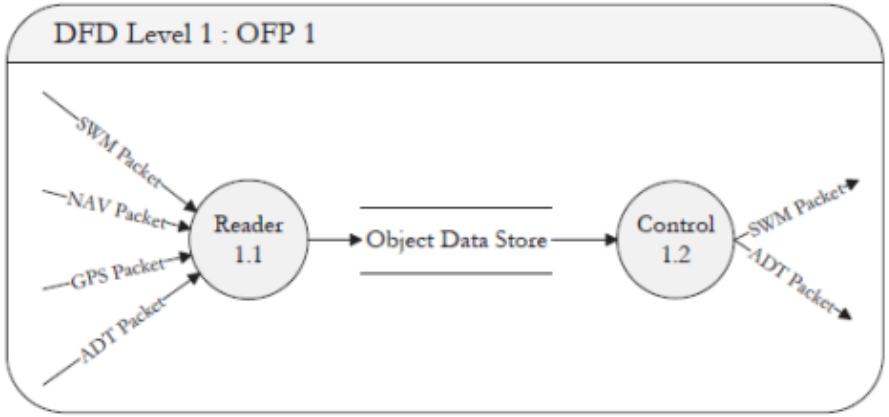
- **Design Recovery**
  - 소스코드와 기존의 가용한 문서 등을 토대로 Design abstraction을 작성
  - 현장/분야 전문가의 경험과 지식을 추가하여 문서를 재 생산

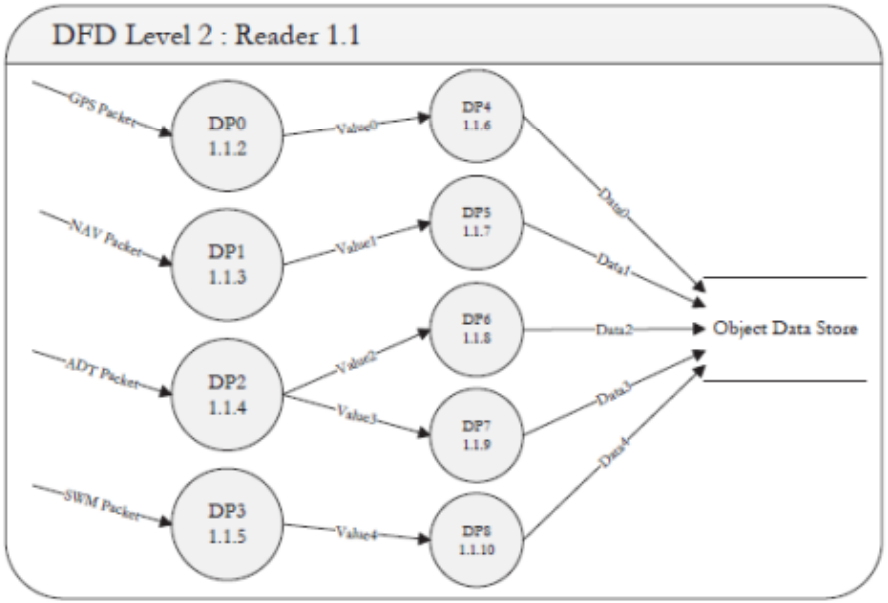
- **DO-178B에서 요구하는 디자인 수준의 내용**
  - Low-level requirements
  - Interface definitions
  - Data flow
  - Control flow

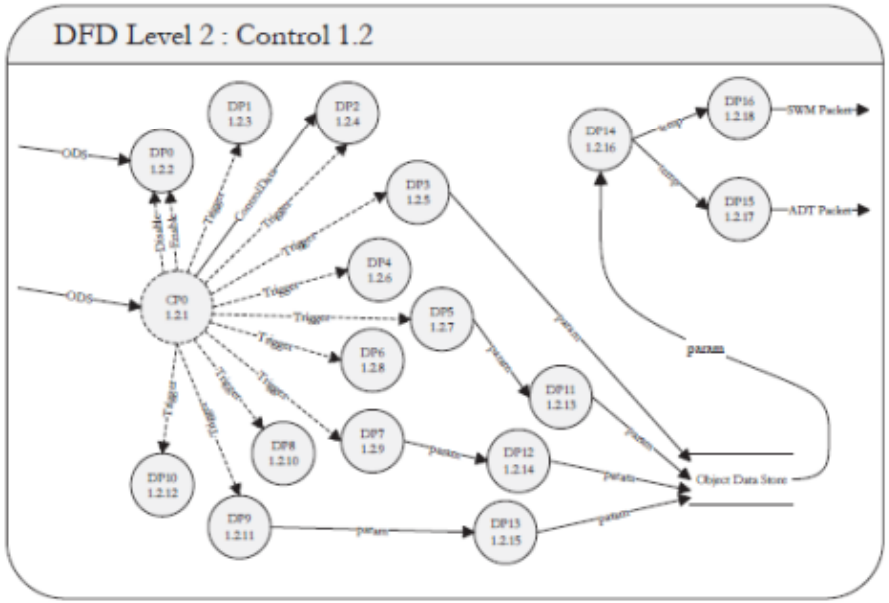
Aio_Close(fu)	void	()	C:\ku it\DrvAio.c
Aio_Init(fu)	int	()	C:\ku it\DrvAio.c
Aio_Read(fu)	int	(int,double *)	C:\ku it\DrvAio.c
Aio_ReadAll(fu)	int	(double *)	C:\ku it\DrvAio.c
Aio_ReadAllAvg(fu)	int	(double *)	C:\ku it\DrvAio.c
Aio_ReadByteAll(fu)	int	(unsigned short *)	C:\ku it\DrvAio.c
AutoLand_Guide(fu)	double	(double,double)	C:\ku it\GnCAutoLand.c
AutoLand_Init(fu)	void	(double,double,double)	C:\ku it\GnCAutoLand.c
CalcCRC(fu)	unsigned short	(unsigned short,unsigned char *,unsigned short)	C:\ku it\crc.c
CalculateBlockCRC32(fu)	unsigned long	(unsigned int,const unsigned char *)	C:\ku it\tGpsReader.c
CRC32Value(fu)	unsigned long	(const int)	C:\ku it\tGpsReader.c
CreateThreads(fu)			
CsToSwmPacket(fu)			
DetectObstacles(fu)			
Dio_ClearChannel(fu)			
Dio_Close(fu)			
Dio_Init(fu)			
Dio_SetChannel(fu)			
Dio_WriteAll(fu)			
ddf(fu)			
df(fu)			
f(fu)			
GetDistanceFromLeg(f			
GetDistanceFromLeg(f			
GetFuturePosition(fu)			
GetFuturePosition(fu)			
GetModeOfPN(fu)			
GetModeOfPN(fu)			
GetPathTarget(fu)			
GetPathTarget(fu)			
GetPushTarget(fu)			
GetPushTarget(fu)			
GnCParams_Get(fu)			
GnCParams_GetWayp			
GnCParams_Init(fu)			
GnCParams_Set(fu)			
GnCParams_SetWayp			
HoldAltitude(fu)			
HoldAutopilot(fu)			
HoldForwardSpeed(fu)			
HoldHeading(fu)			
HoldSideSpeed(fu)	double	(double,double,double,double,int)	C:\ku it\GnCAutoPilot.c
InitAutopilot(fu)	void	()	C:\ku it\GnCAutoPilot.c
InitResources(fu)	int	()	C:\ku it\main.c
IsBodyValid(fu)	int	(const unsigned char [])	C:\ku it\tGpsReader.c
IsBodyValid(fu)	int	(const unsigned char [])	C:\ku it\tGpsReader.c
IsChecksumValid(fu)	int	(const unsigned char [])	C:\ku it\tSwmReader.c
IsChecksumValid(fu)	int	(const unsigned char [],const int)	C:\ku it\tNavReader.c
IsCrcValid(fu)	int	(const unsigned char [],const int)	C:\ku it\tAdtReader.c
IsHeaderValid(fu)	int	(const unsigned char [])	C:\ku it\tAdtReader.c
IsHeaderValid(fu)	int	(const unsigned char [])	C:\ku it\tGpsReader.c
IsHeaderValid(fu)	int	(const unsigned char [])	C:\ku it\tGpsReader.c

Interface Definitions









Data Flow Diagrams

요구사항 수준의 역공학

- **요구사항 수준의 문서**
  - Requirements Specification
    - Functional requirements
    - Non-functional requirements
    - Assumptions

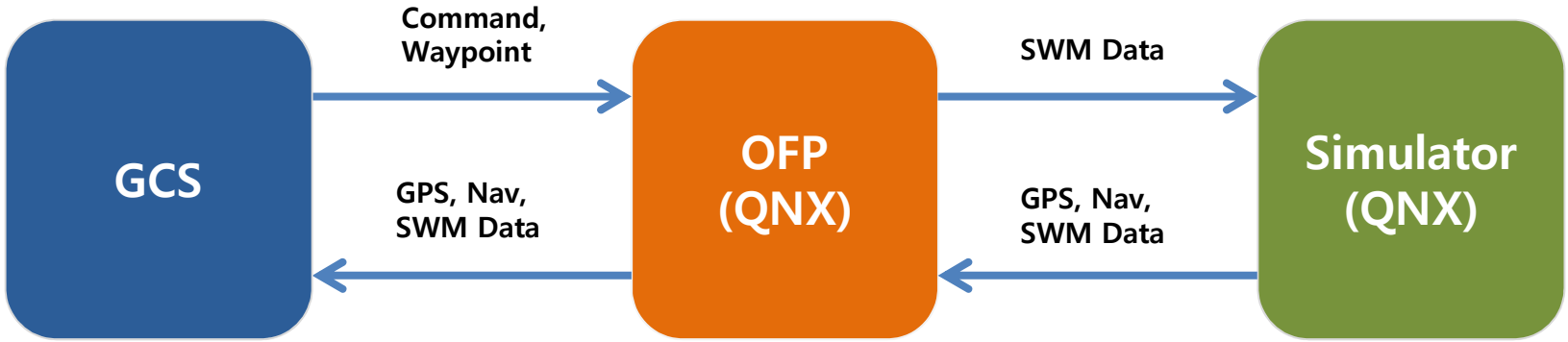
SOFTWARE TESTING
(SW 테스팅)

기능 테스트

- **OFP 코드에 대한 기능 테스트(Functional Testing)**
  - OFP가 디자인/요구사항 명세서에 따라 동작하는 가를 검사
  - 역공학을 통해 복원된 문서들을 활용하여, 테스트 요구사항(test requirements)과 테스트 케이스(test cases)를 개발
  - 많은 부분이 수작업으로 진행됨

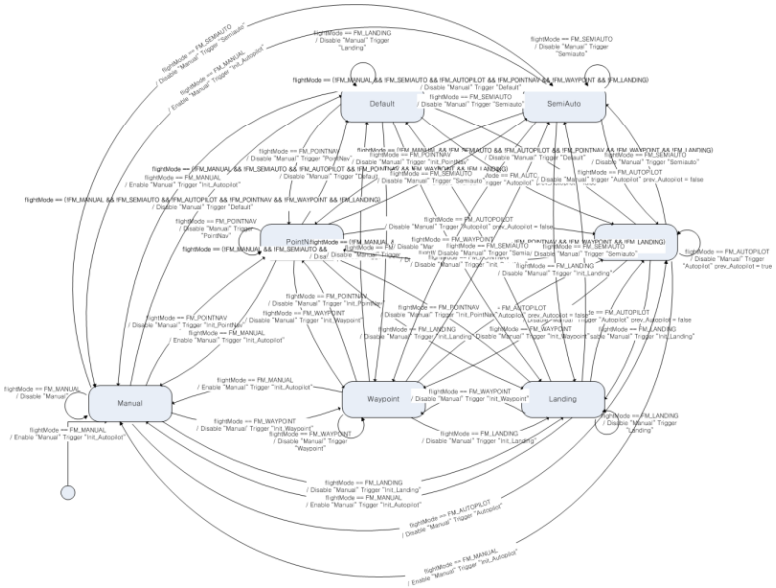
- **적용 가능한 기능 테스트 기법들**
  - Brute-forced test
  - Pairwise test
  - Category-partitioning test
  - **Boundary value test**
  - **(FSM) Model-based test**
  - 그 외 다수

- 기능 테스트를 위한 환경 설정
  - HILS(Hardware-in-the-loop System)
    - Real-time embedded software인 OFP를 Testing하기 위한 환경
    - OFP와 GCS, 하드웨어 Simulator로 구성
    - QNX를 이용하여 목표 환경을 구축

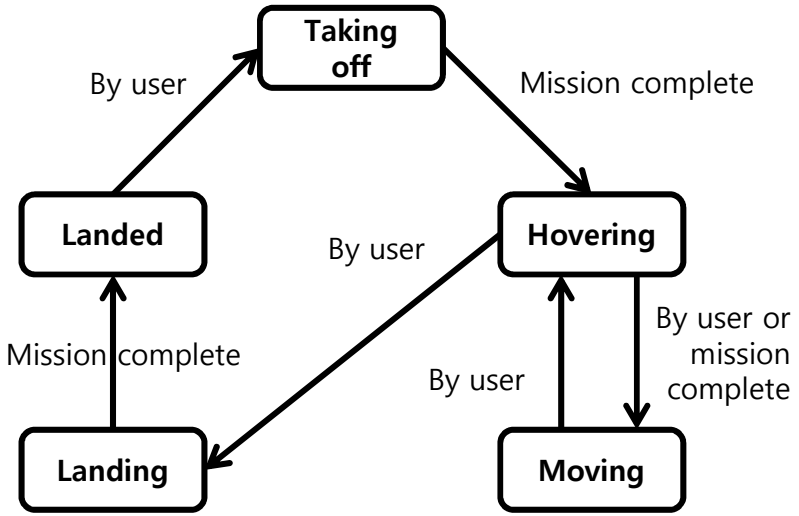


기능 테스트의 구체적인 내용 및 결과

- 상태 전이에 기반한 기능 테스트
  - 역공학으로 복원된 정보와 기존 문서의 정보를 합하여 테스트 케이스 생성
    - 역공학으로 복원된 Controller의 상태 정보
    - 기존 문서에 존재하는 자동비행 모드에서의 상태 변화

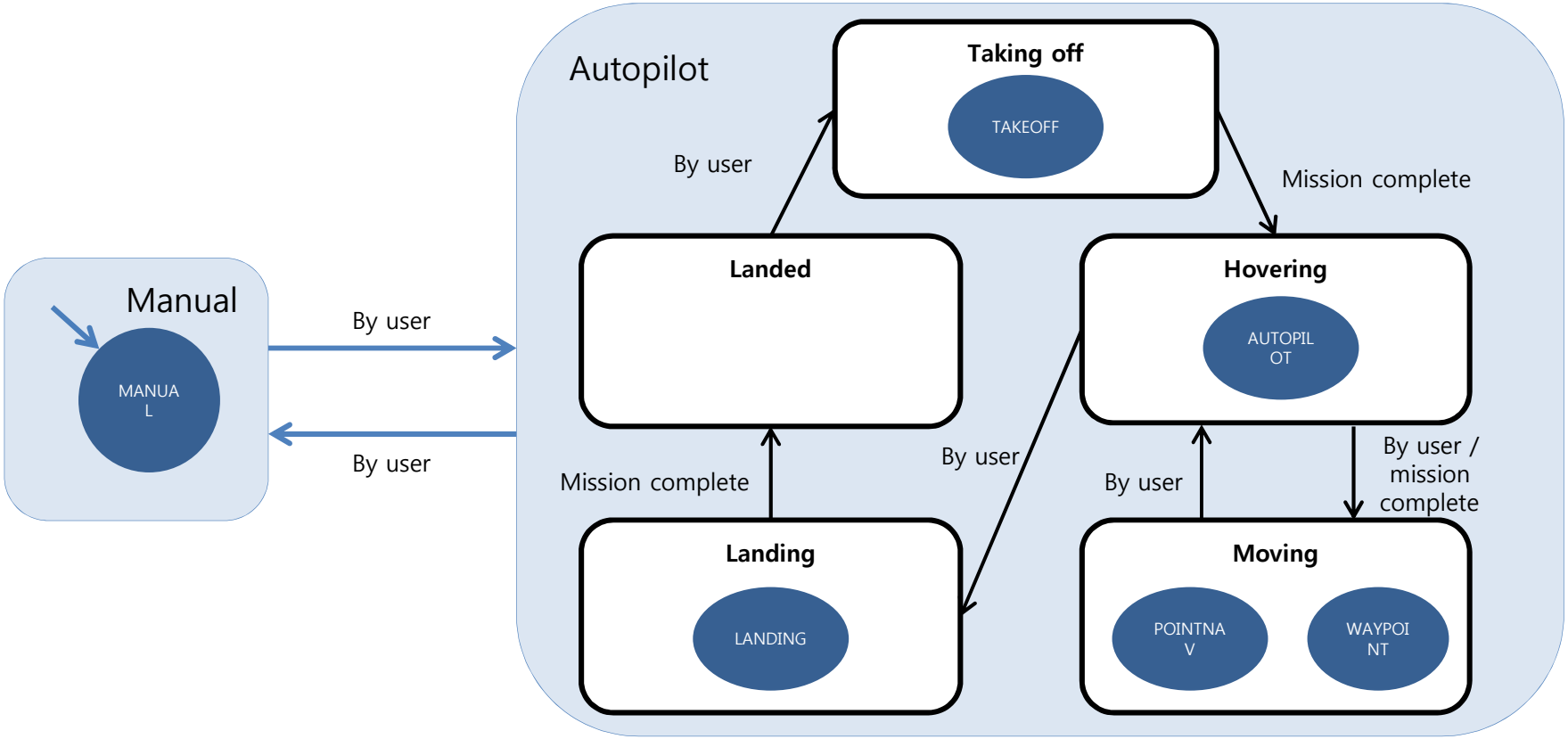


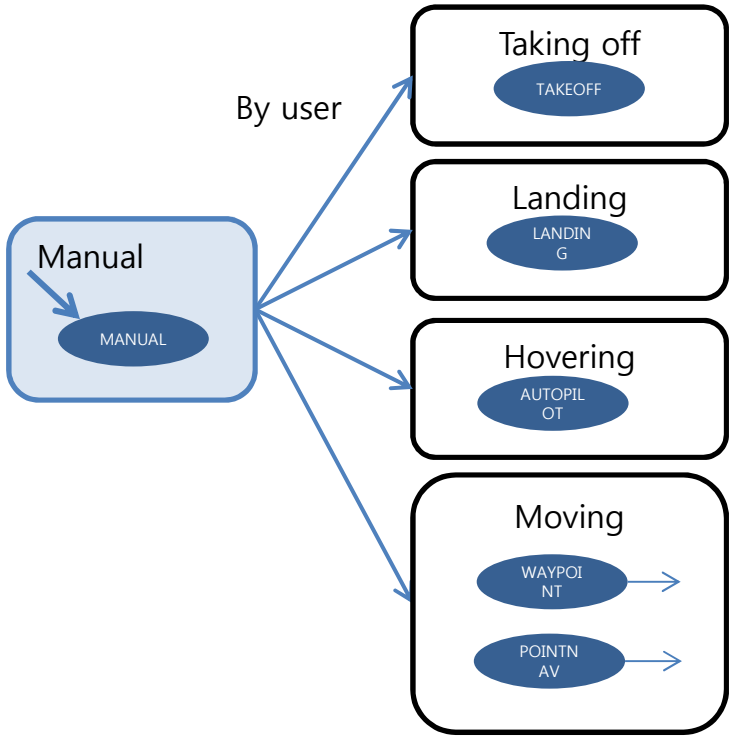
역공학으로 복원된 정보

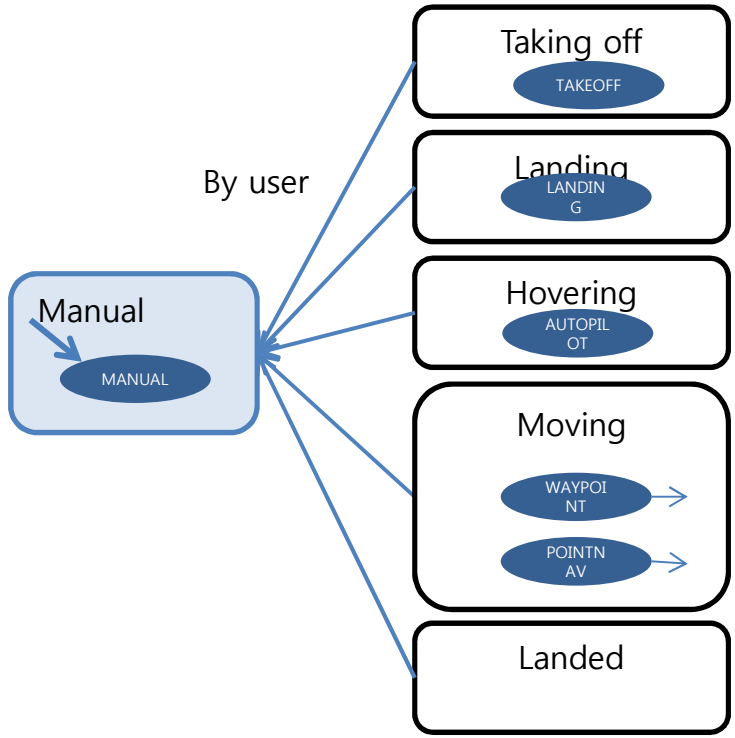


기존 문서에 존재하는 정보

- **생성된 정보를 바탕으로 테스트 케이스 생성**
  - 미션완료 및 사용자에 의한 모든 상태변화에 대한 테스트 케이스 생성



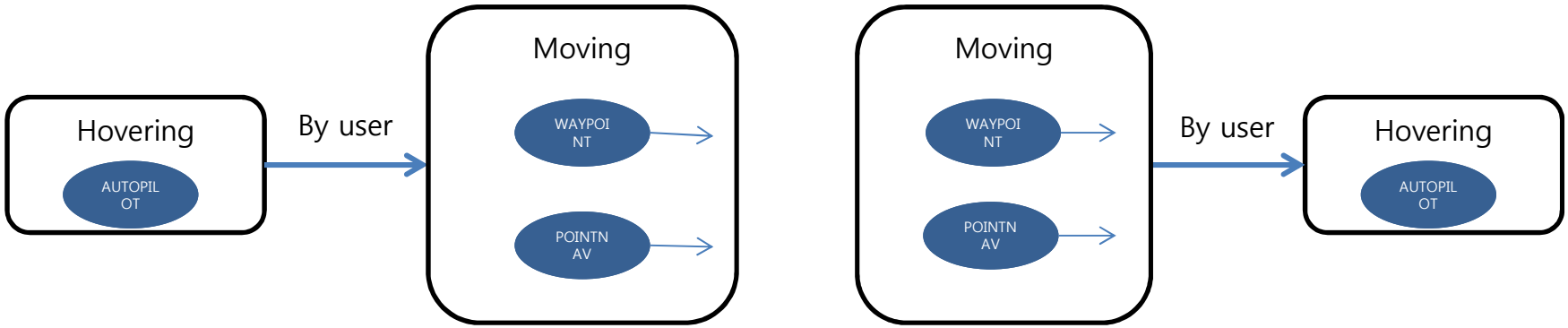




ID	flightMode(prev)	flightMode(next)
TC_S00	FM_MANUAL	FM_TAKEOFF
TC_S01	FM_MANUAL	FM_LANDING
TC_S02	FM_MANUAL	FM_AUTOPILOT
TC_S03	FM_MANUAL	FM_POINTNAV
TC_S04	FM_MANUAL	FM_WAYPOINT

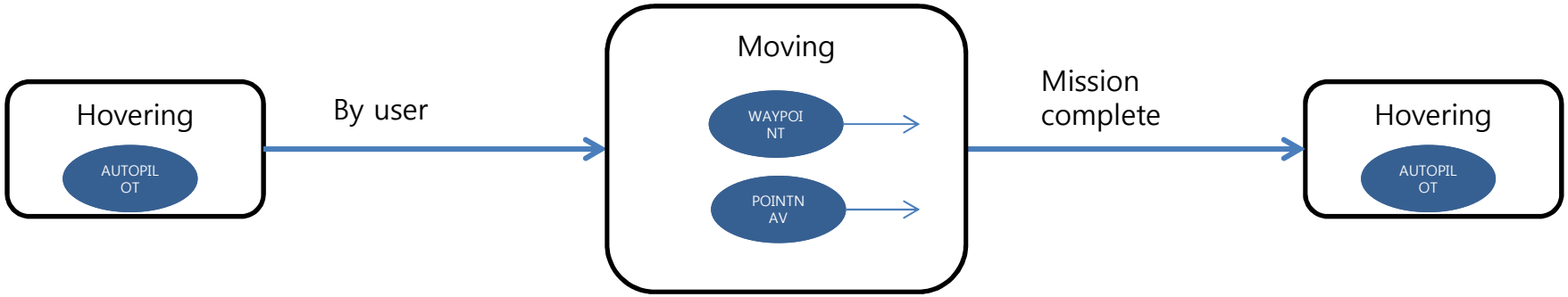
ID	flightMode(prev)	flightMode(next)
TC_S06	FM_TAKEOFF	FM_MANUAL
TC_S07	FM_LANDING	FM_MANUAL
TC_S08	FM_AUTOPILOT	FM_MANUAL
TC_S09	FM_POINTNAV	FM_MANUAL
TC_S10	FM_WAYPOINT	FM_MANUAL

Landed와 대응되는 state가 없기 때문에 test cases 생성 불가

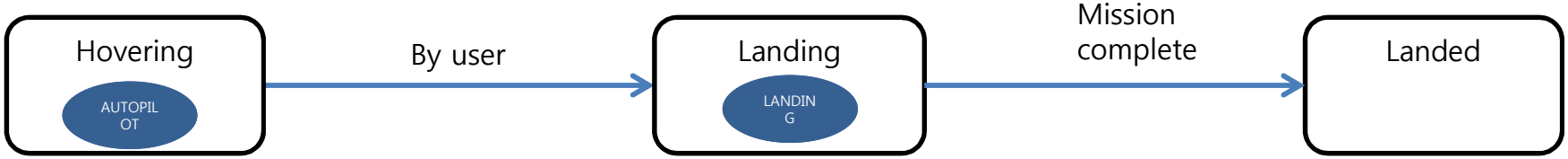


ID	flightMode(prev)	flightMode(next)
TC_S11	FM_AUTOPILOT	FM_POINTNAV
TC_S12	FM_AUTOPILOT	FM_WAYPOINT
TC_S13	FM_AUTOPILOT	FM_WAYPOINT → FM_POINTNAV

ID	flightMode(prev)	flightMode(next)
TC_S14	FM_POINTNAV	FM_AUTOPILOT
TC_S15	FM_WAYPOINT	FM_AUTOPILOT



ID	flightMode(prev)	flightMode(next)	flightMode(next)
TC_S16	FM_AUTOPILOT	FM_WAYPOINT	FM_AUTOPILOT
TC_S17	FM_AUTOPILOT	FM_POINTNAV	FM_AUTOPILOT



Landed와 대응되는 state가 없기 때문에 test cases 생성 불가

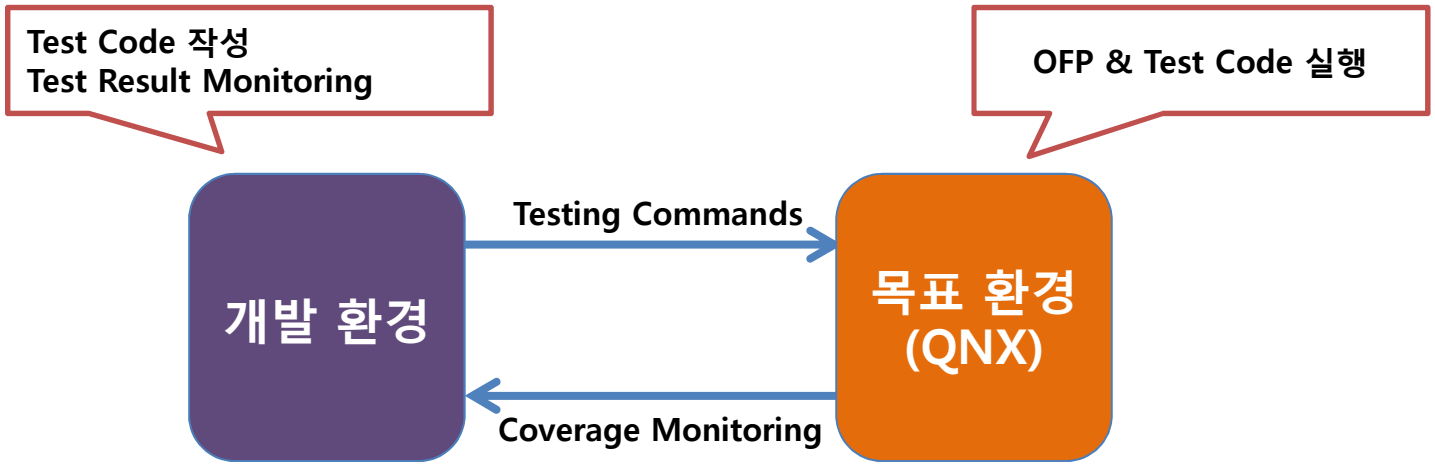
구조 테스트

- **OFP 코드에 대한 구조 테스트(Structural Testing)**
  - OFP 코드의 컨트롤 구조를 중심으로 테스트를 수행
  - 다양한 구조적인 속성(Coverage Criteria)을 만족하는 테스트 케이스를 개발
  - 많은 부분을 **자동화 도구**를 이용하여 진행

- **적용 가능한 구조적인 속성 (Structural Test Coverage Criteria)**
  - Statement
  - Branch / Condition / Decision
  - **MC/DC (Modified Condition and Decision Coverage)**

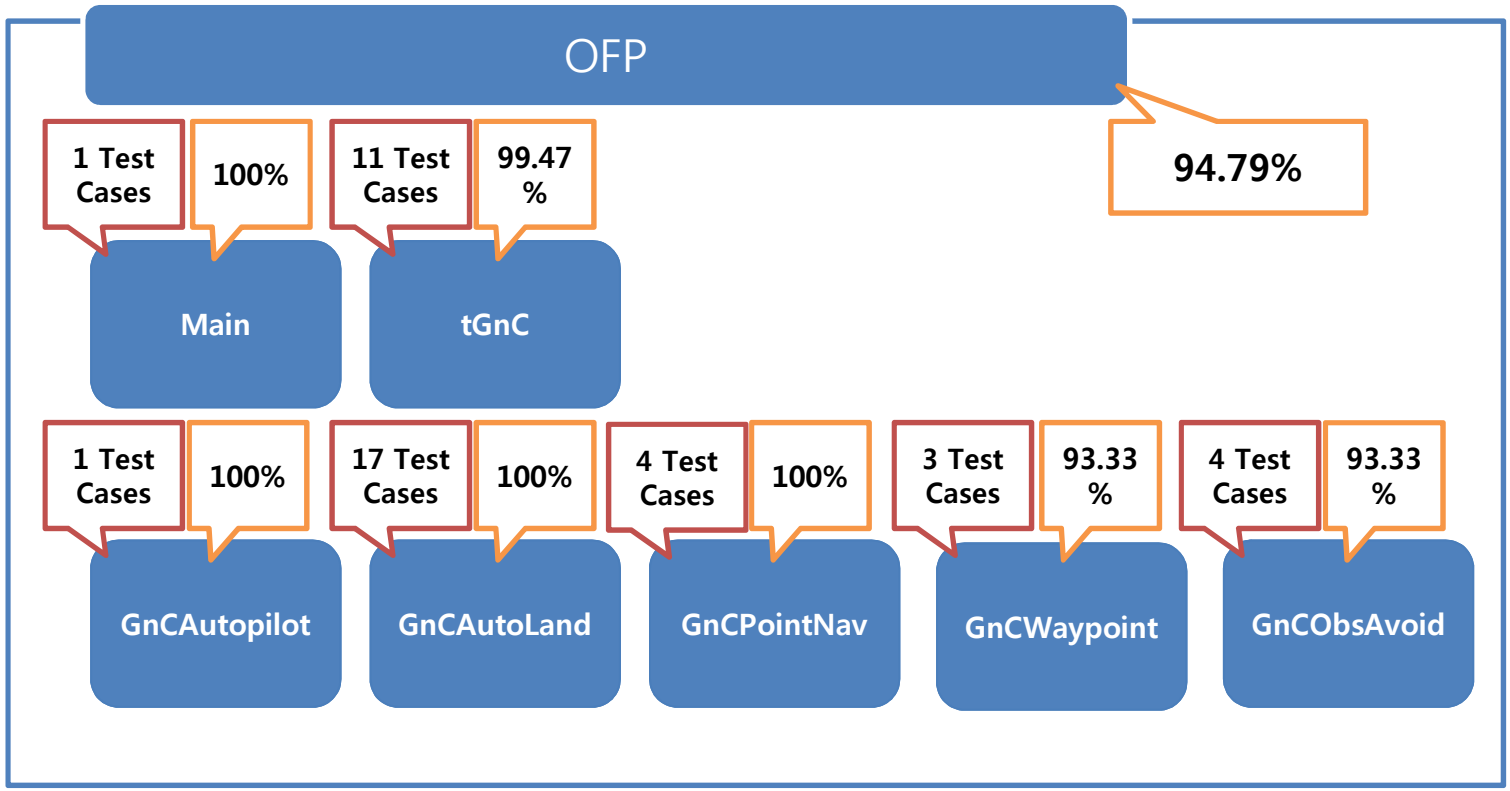
- **적용 가능한 기능 테스트 자동화 도구**
  - Code Scroll
  - LDRA
  - Gcov

- 구조 테스트를 위한 환경 설정
  - HILS(Hardware-in-the-loop System)
  - 개발 환경과 목표 환경을 각각 구성하여 Testing 수행
    - Test code 작성 : in 개발 환경(Windows)
    - Run/Debug : in 목표 환경(QNX)



구조 테스트의 구체적인 내용 및 결과

- **구조 테스트 수행**
  - OFP를 여러 개의 모듈로 나누어 Unit Level의 구조 테스트 수행
  - 각 모듈을 검사하기 위한 Test Cases 개발
  - Statement Coverage 측정 (94.79%)



정적 코드분석

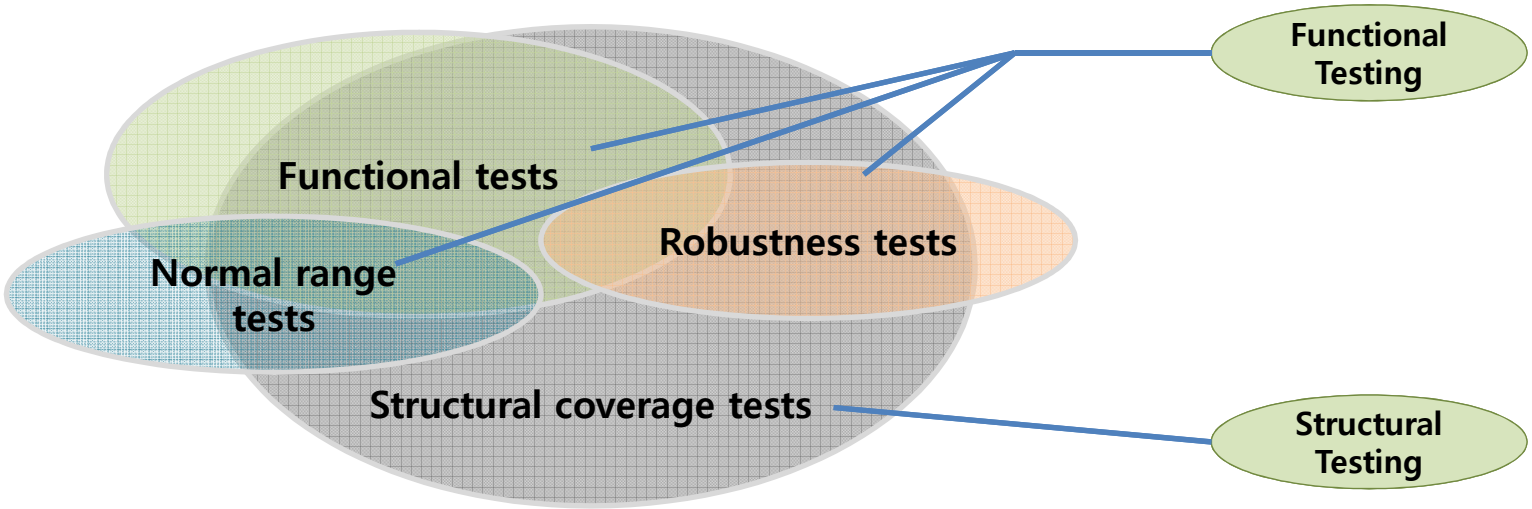
- **OFP 코드에 대한 정적 코드분석(Static Analysis)**
  - Checklist 또는 도구를 활용한 code analysis
  - Software를 실행하지 않고 검사
    - 별도의 환경구축이 필요하지 않음

- **적용 가능한 정적 코드분석 도구**
  - Coverity Prevent
  - Sparrow
  - cppcheck
  - 기타

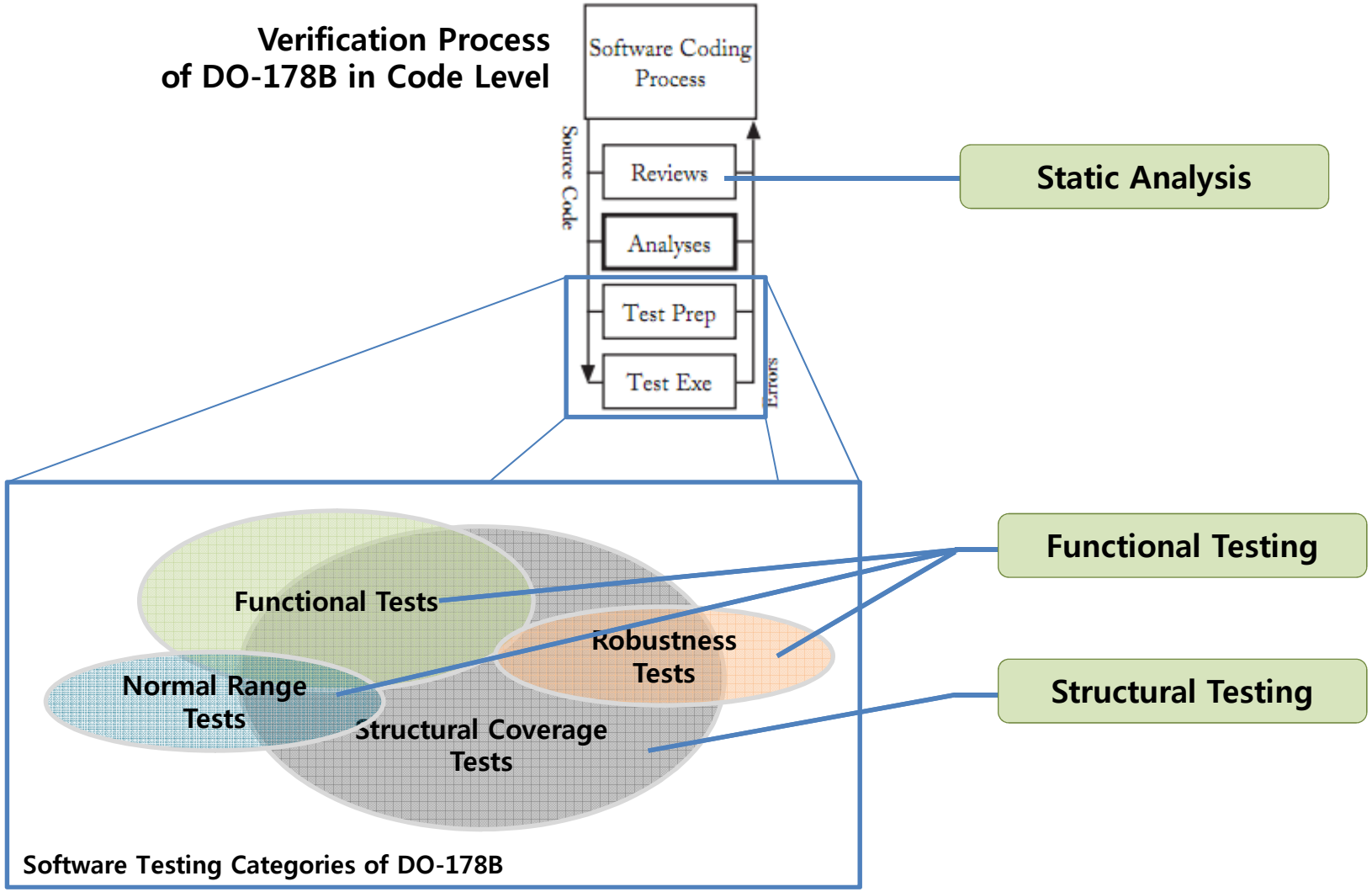
- **Checklist**
  - Code syntax , programming rule , 불필요한 코드 검사
  - Algorithms 검사

참고 (DO-178B SW Testing)

- **4개의 Category**
  - Functional tests
  - Normal range tests
  - Robustness tests
  - Structural coverage tests



Verification Process of DO-178B in Code Level



맺음말

맺음말

- **UAV & SW Fusion Research Center (UAV-SW)**
  - 무인항공기/지상제어시스템 全 SW 체계 개발
    - 응용SW ~ 운영체제/미들웨어
    - OFP ~ GCS
- **SW 全 개발 Lifecycle에 대한 SW Verification & Validation 수행**
  - 부족한 문서를 보완하기 위해, Reverse Engineering 실시
- **DO-178B 등 인증표준 준용**
  - Level A
  - 기능시험(Functional Testing)과 구조시험(Structural Testing)을 병행 수행
  - 코드분석(Static Analysis) 기법 적용
- **정형기법 적용**
  - OFP 정형 모델 개발
  - 정형검증 수행

- 국방부 u-실험사업 사업 제안
  - *Ubiquitous* IT 중심 사업의 SW 안전성 확인 체계 수립
    - 동기
      - Ubiquitous IT 기술을 도입할수록 더 많은 SW 안전성 문제가 발생
      - 각 사업 별로, SW 안전성을 검증/인증 할 수 있는 방법의 先決 필요
    - 주요 내용
      - 추진 예정 사업 별로, 독립적인 과제/기관에서
      - SW 안전성 확보 방안 수립 (事前)
      - SW 안전성 검증/인증 방안 수립 (事後)