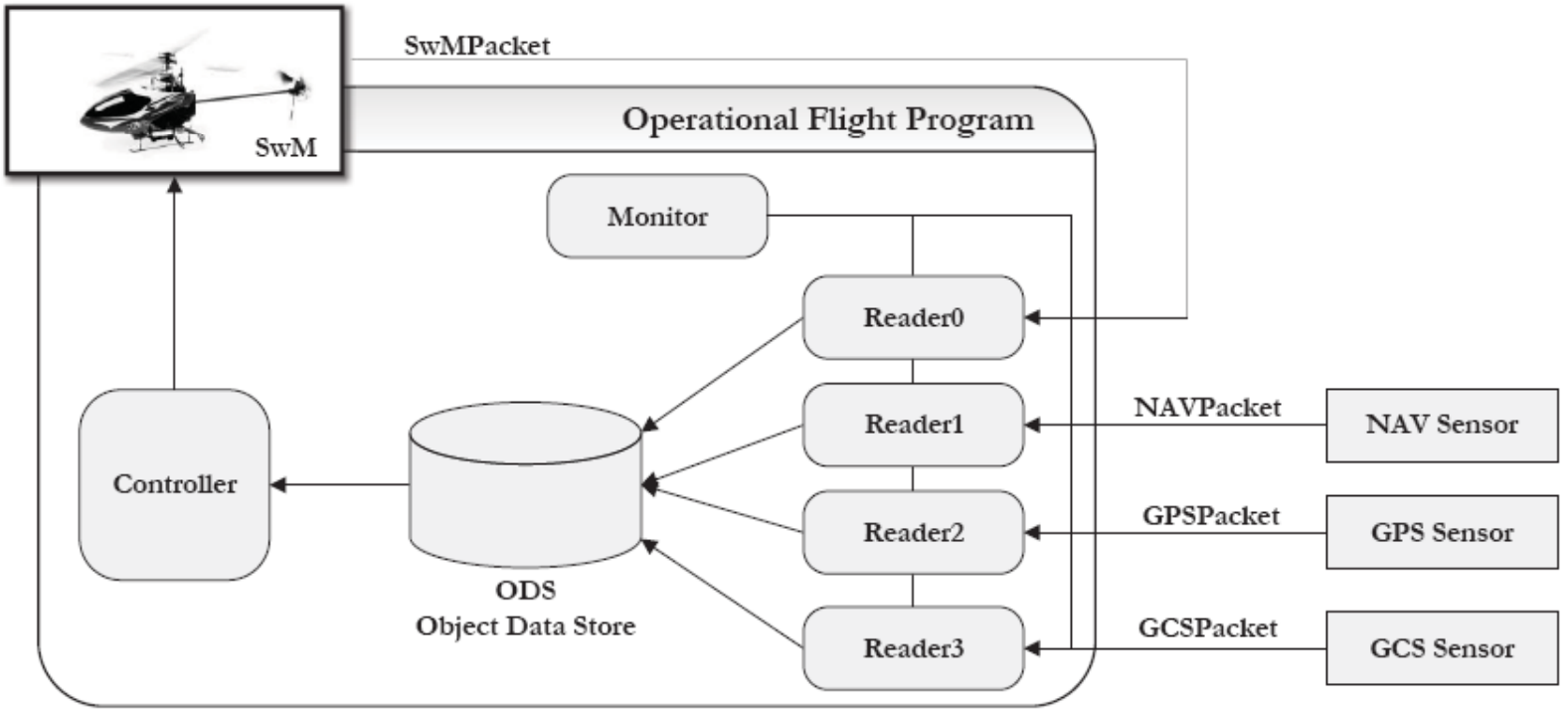
ASEA 2011
Jeju Island, Korea
2011.12.08~10

Systematic Verification of Operational Flight Program through Reverse Engineering

JUNBEOM YOO
KONKUK UNIVERSITY

OFP(Operational Flight Program) for UAVs

- **Real-Time Embedded Control Software for UAV**
  - 6 Processes (1 Controller, 1 Monitor, 4 Reader) , 3 Shared Data Area (ODS)
  - Working on TMO (Time Triggered Message)

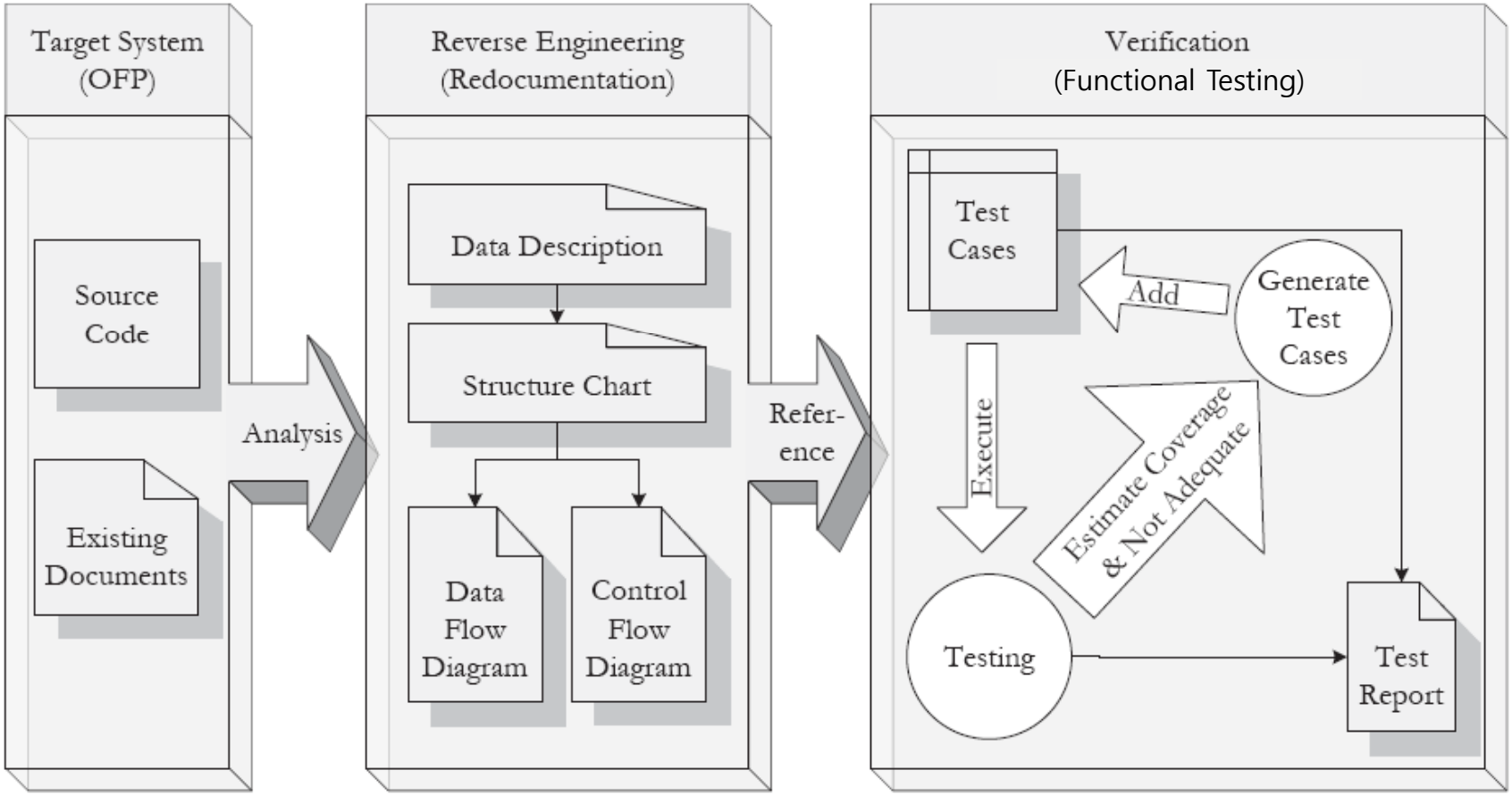


UAVs under Development

HeliScope

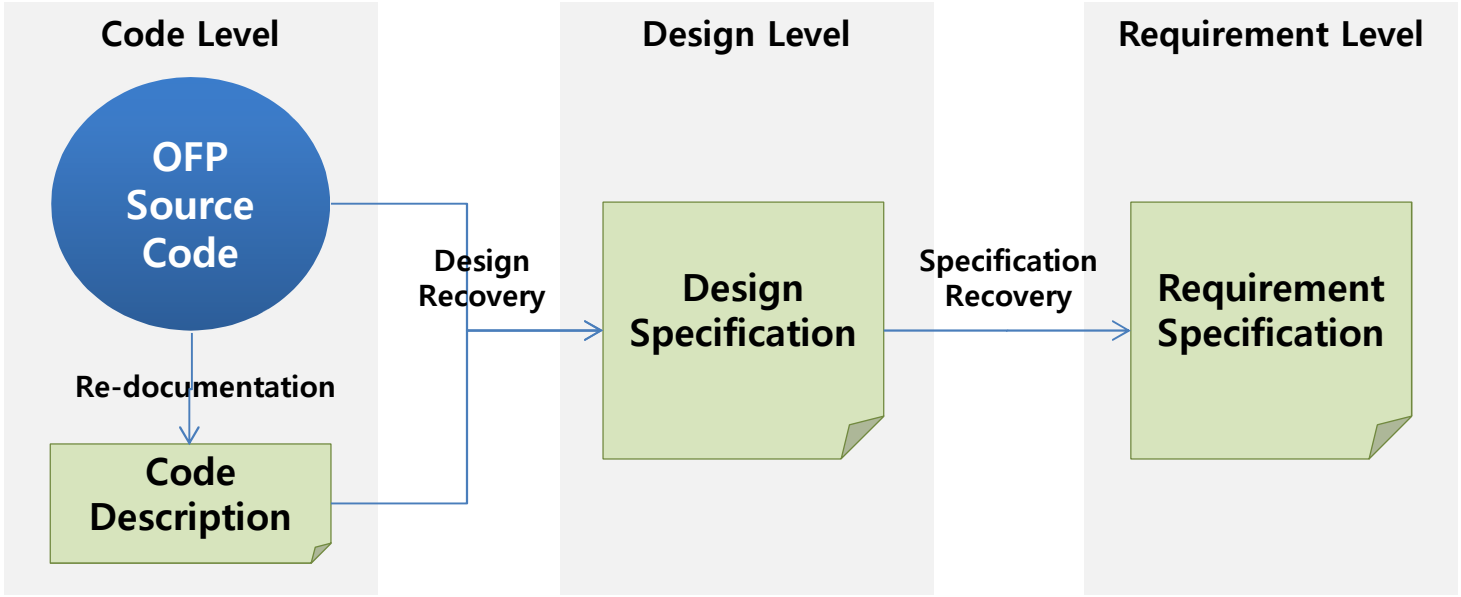
QuadScope & Mobile GCS

Testing Plan for the OFP



Reverse Engineering for the OPF

- **Re-documentation and Recovery of designs**
  - Recovery of missed design documents
  - Document standardization
  - Resources for V&V activities
  - Conducted manually

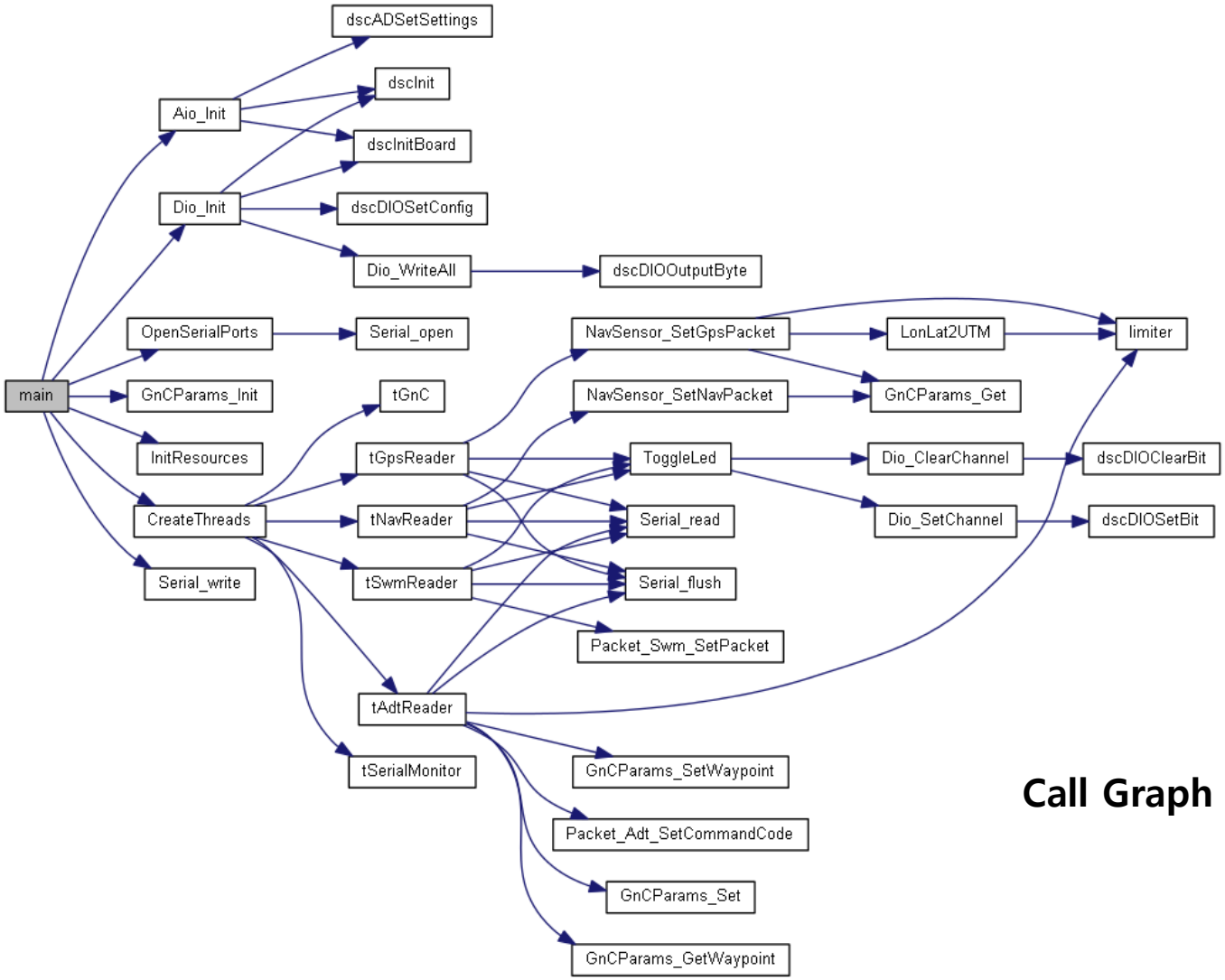


Reverse Engineering – Code Level

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

Reverse Engineering – Code Level



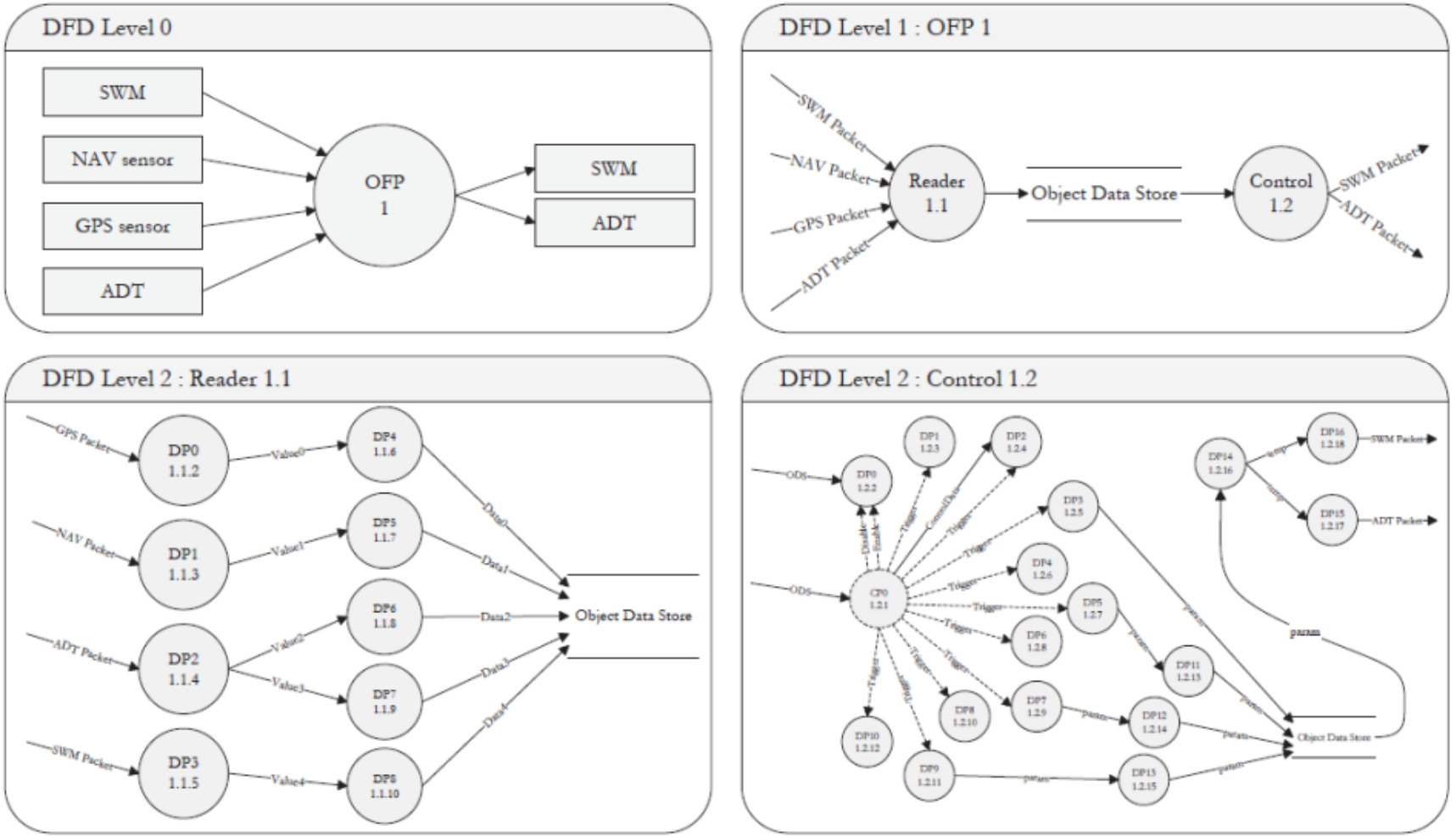
Call Graph

Reverse Engineering – Design Level

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)	int	()	C:\ku it\main.c
CsToSwmPacket(fu)	unsigned int	(double,double,double,double,unsigned char [])	C:\ku it\tGnC.c
DetectObstacles(fu)	int	(double,double,double)	C:\ku it\GnCObsAvoid.c
Dio_ClearChannel(fu)	int	(BYTE,BYTE)	C:\ku it\DrvDio.c
Dio_Close(fu)	void	()	C:\ku it\DrvDio.c
Dio_Init(fu)	int	()	C:\ku it\DrvDio.c
Dio_SetChannel(fu)	int	(BYTE,BYTE)	C:\ku it\DrvDio.c
Dio_WriteAll(fu)	int	(BYTE,BYTE)	C:\ku it\DrvDio.c
ddf(fu)	double	(double)	C:\ku it\GnCAutoLand.c
df(fu)	double	(double)	C:\ku it\GnCAutoLand.c
f(fu)	double	(double)	C:\ku it\GnCAutoLand.c
GetDistanceFromLeg(fu)	double	(int,double,double)	C:\ku it\GnCObsAvoid.c
GetDistanceFromLeg(fu)	double	(int,double,double)	C:\ku it\GnCPointNav.c
GetFuturePosition(fu)	void	(double,double *,double *)	C:\ku it\GnCObsAvoid.c
GetFuturePosition(fu)	void	(double,double *,double *)	C:\ku it\GnCPointNav.c
GetModeOfPN(fu)	int	(int,double,double)	C:\ku it\GnCObsAvoid.c
GetModeOfPN(fu)	int	(int,double,double)	C:\ku it\GnCPointNav.c
GetPathTarget(fu)	void	(int,double,double,double *,double *)	C:\ku it\GnCObsAvoid.c
GetPathTarget(fu)	void	(int,double,double,double *,double *)	C:\ku it\GnCPointNav.c
GetPushTarget(fu)	void	(int,double,double,double *,double *)	C:\ku it\GnCObsAvoid.c

Interface Definitions

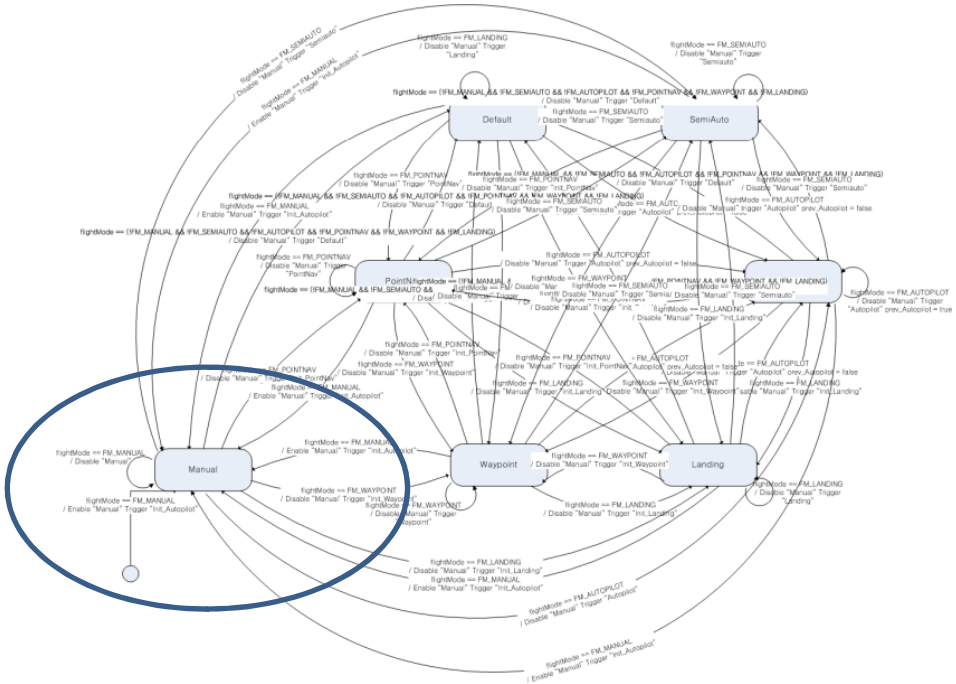
Reverse Engineering – Design Level



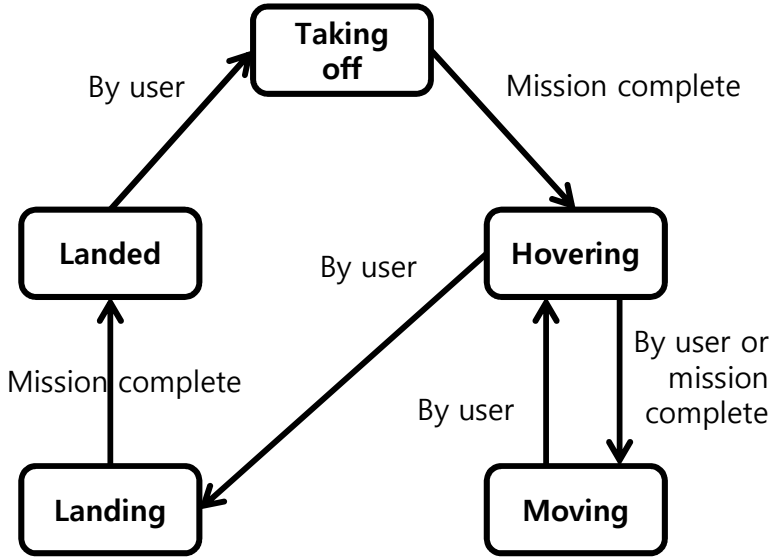
Data Flow Diagrams

Software Testing

- **FSM-based Testing**
  - Using recovered information to develop test cases
    - State transitions of the controller in the OFP

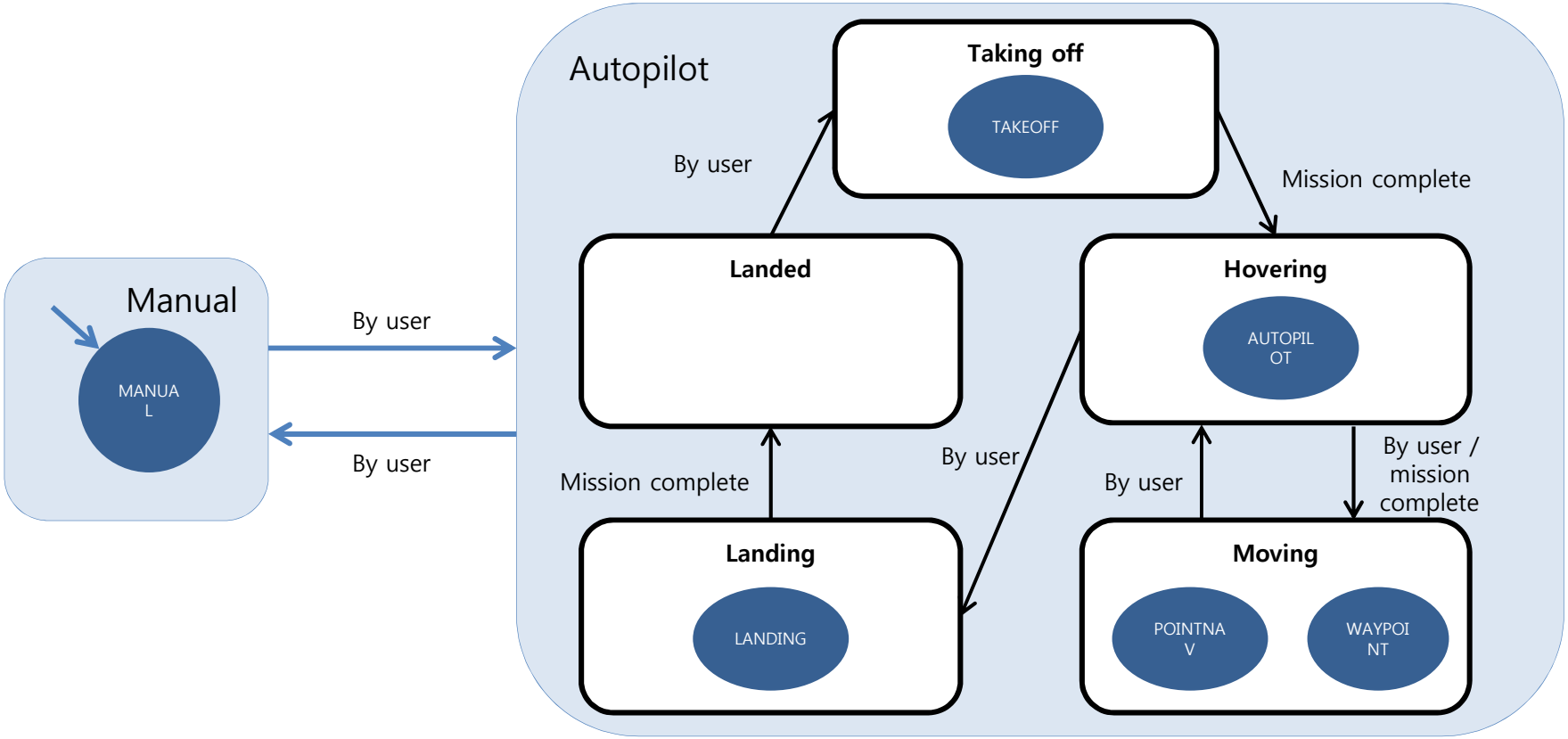


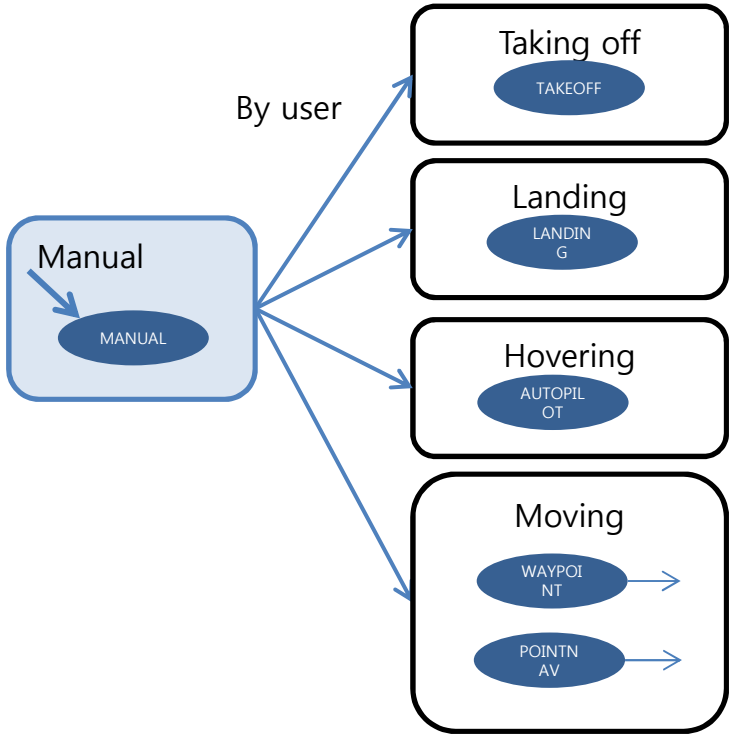
Recovered Information

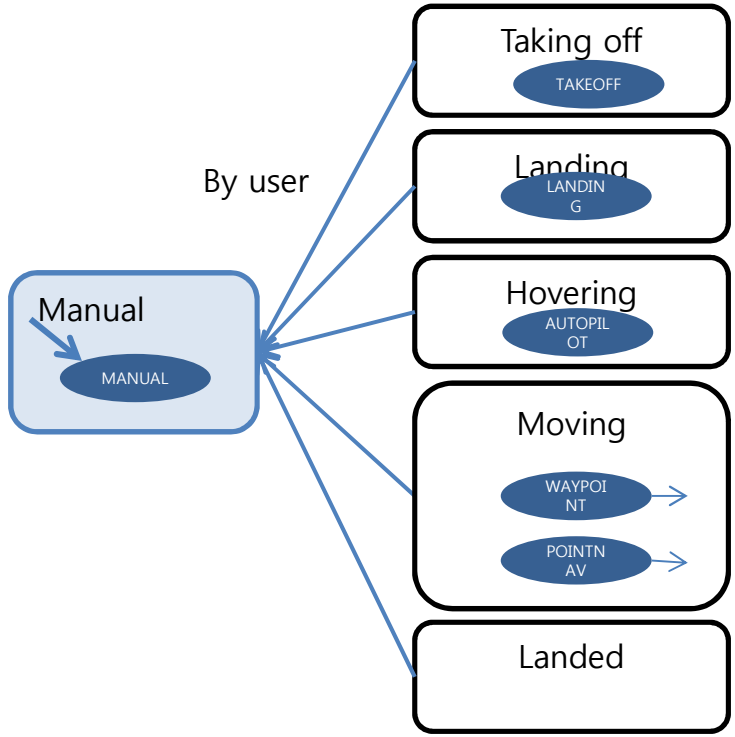


Existing Information

- **Modified FSM**
  - Manual ↔ Autopilot



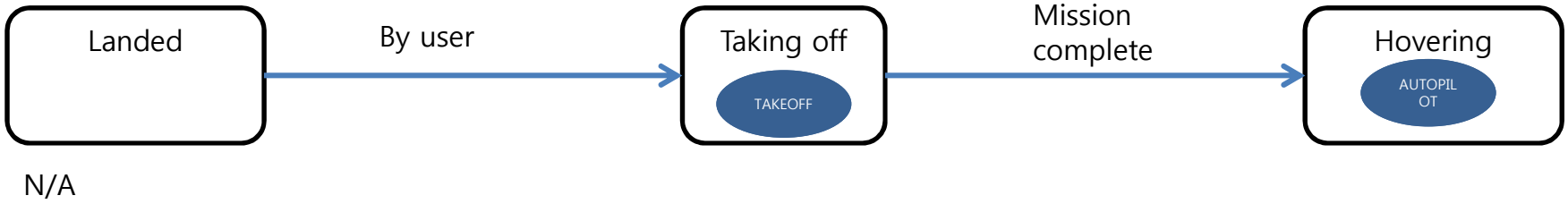




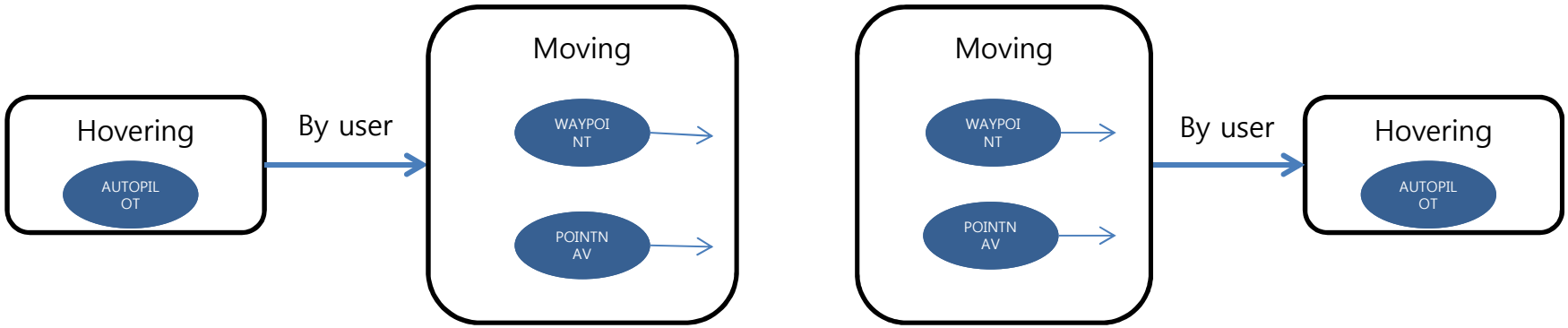
Test Cases

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

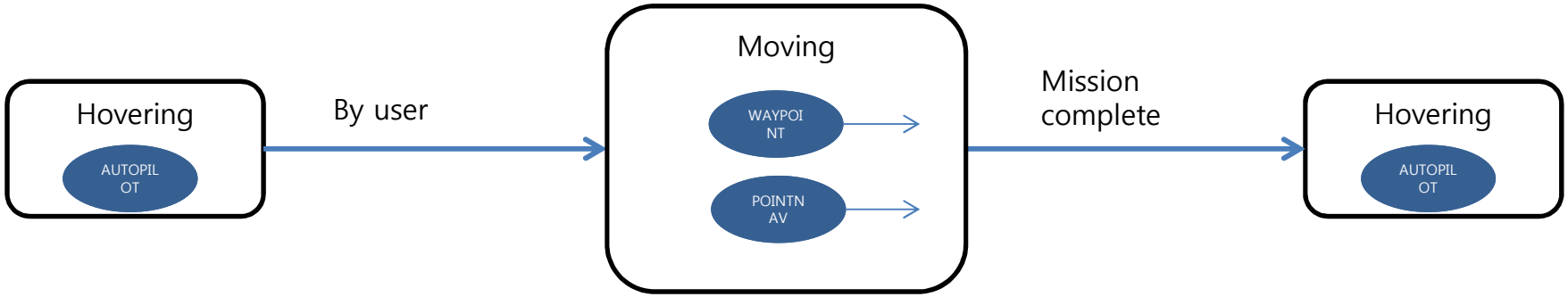


N/A

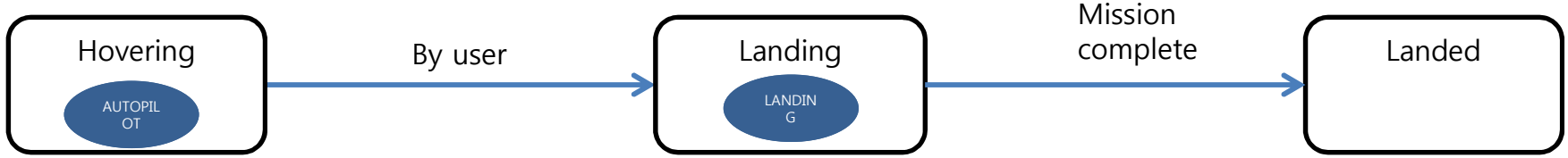


ID	flightMode(prev)	flightMode(next)
TC_S11	FM_AUTOPILOT	FM_POINTNAV
TC_S12	FM_AUTOPILOT	FM_WAYPOINT
TC_S13	FM_AUTOPILOT	FM_WAYPOINT → FM_POINTNAV

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



N/A

Conclusion and Future Work

- **Testing the UAV's OFP**
  - Applied reverse engineering techniques for recovering missed documents
  - Used the recovered information to develop functional test cases

- **Test execution**
  - Planning to execute the developed test cases on the HILS(Hardware-In-the-Loop Simulation) environment