

다중 무인체계 유무인 협동 시뮬레이터 개발



파란학기 팀: AIE

지도교수: 권 용 진 교수

목차

- 연구 일정

- 연구 진행 현황

- 연구 내용

- ✓ 전체 시뮬레이터 구조
- ✓ SW간 **ICD** 정의
- ✓ **Command** Software
- ✓ **Control** Software
- ✓ **Simulation** Software
- ✓ 구현 **Demo**

- 연구 산출물

- 향후 연구 계획

1. 연구 일정

1. 연구 일정

❖ 연구 일정 계획

연구 개발 항목	추진 일정 (개월)											비 고
	12	1	2	3	4	5	6	7	8	9	10	
<div>▪ 단일 객체 모델 개발</div> <div>✓ S/W 구조 모듈화</div> <div>✓ S/W 데이터 송·수신을 위한 I/O 인터페이스 구축</div>	<div></div>											
<div>▪ 복수 객체 모델 확장</div> <div>✓ 해상/지상/공중 시뮬레이션 환경 모델링</div> <div>✓ 복수 객체 핸들링을 위한 체계적 구조 구축</div>	<div></div>											
<div>▪ 주요 모듈 별 구조화 설계</div> <div>✓ 시뮬레이션 S/W의 구조 확장</div> <div>✓ AI 학습이 가능한 모듈의 상세 설계</div>	<div></div>											
<div>▪ 시나리오에 따른 시뮬레이터 검증 및 평가</div>	<div></div>											
<div>▪ 군집 임무 계획 알고리즘 연구</div>	<div></div>											

2022. 06. 07

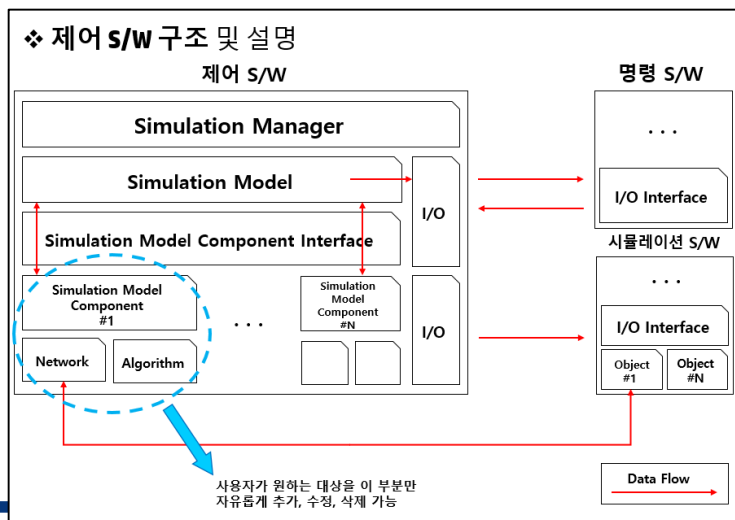
2. 연구 진행 현황

❖ 연구 진행 현황

❖ 연구 진행 현황

- » 다기종 시뮬레이션 운영이 가능하고 변경 가능성과 확장 가능성을 가진 소프트웨어 모듈화 구조 설계
 - ✓ 제어 S/W를 중심으로 명령 S/W, 시뮬레이션 S/W의 유연한 구조 설계로 추후 모델 추가 및 수정 과정에서 **Dependency** 감소
- » 모듈화 된 S/W 사이의 인터페이스 ICD(Interface Control Document) 정의
 - ✓ 추가, 수정을 고려한 유연한 인터페이스 ICD 정의
- » 해양 환경 모델링
 - ✓ Raw Height Map을 이용한 진해 앞바다 환경 모델링

◆ 제어 S/W를 중심으로 한 소프트웨어 구조



◆ S/W간 인터페이스 ICD

❖ **S/W 간 ICD 정의** - 실시간 통신 데이터 ICD 예시 - From Unity To Control (USV)

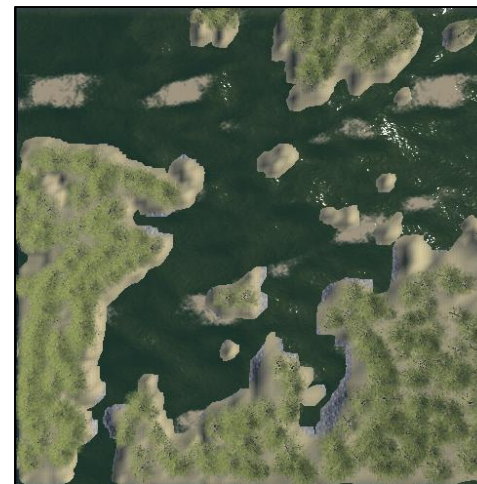
Byte	Data Information	Type
0 ~ 3	현재 위치 x	float
4 ~ 7	현재 위치 y	float
8 ~ 11	현재 위치 z	float
12~15	Euler 각 x	float
16~ 19	Euler 각 y	float
20 ~ 23	Euler 각 z	float
24 ~ 27	감지된 장애물 수	Int
28 ~ 31	감지된 장애물의 x 좌표	float[]
32 ~ 71	감지된 장애물의 y 좌표	float[]
72 ~ 111	감지된 장애물의 z 좌표	float[]
112 ~ 151	감지된 장애물과의 거리	float[]

공통 데이터

USV 전용 데이터

새로운 플롯을 추가 시 해당 파트만 수정하여 기반 클래스 상속

◆ 해양 환경 모델링

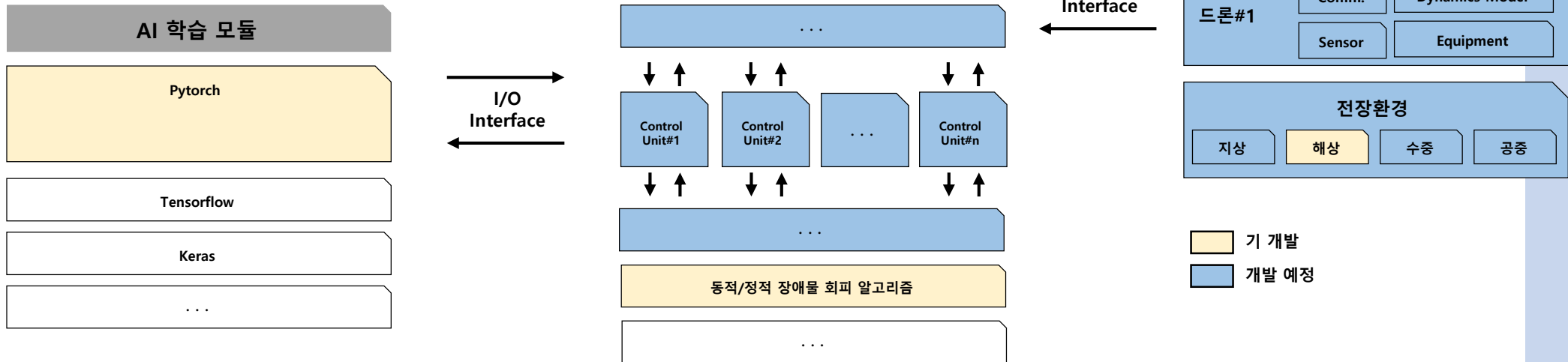


3. 연구 내용

3. 연구 내용

❖ 전체 시뮬레이터 구조도

- S/W를 기능별로 모듈화, **추가/수정/제거**가 용이한 구조로 설계
- 명령 S/W는 사용자가 원하는 시나리오 및 객체 설정 기능을 통해 시뮬레이션 환경 설정
- 제어 S/W는 시뮬레이션 과정에 필요한 실질적인 알고리즘이 수행되는 곳으로, 모듈화된 S/W들 중앙에서 **시뮬레이션을 제어**
- 시뮬레이션 S/W는 제어 S/W와의 I/O 인터페이스를 통해 받은 데이터를 기반으로 시뮬레이션 **수행 및 가시화**
- AI 학습 모듈과 제어 S/W 연동을 통해 다양한 딥러닝/강화 학습 알고리즘 트레이닝 수행



❖ S/W 간 ICD 정의 - Effective ICD

ICD

Effective ICD has to include..

- ✓ Interface 및 Interface type에 대한 설명
- ✓ 시스템 간에 전달되는 명령 및 매개 변수에 대한 설명
- ✓ 시스템이 하위 시스템, 외부 시스템과 통신하는 방법 설명
- ✓ 시스템이 외부 명령을 해석하는 방법과 예상되는 응답 및 동작에 대한 설명

순번	필드 명	의미
1	ID	Data Packet 번호
2	Interface	통신 프로토콜 종류
3	Struct_name	Struct 이름
4	Struct_comment	Struct 설명
5	field_cnt	Struct 내 field 개수
6	field_number	Struct 내 filed 순번
7	field_name	Field 이름
8	field_type	Filed 데이터 타입
9	array_size	Filed 배열 사이즈
10	array_byte	Filed byte

- 시스템 뒤의 비즈니스 논리나 기술적 측면을 알지 않고도 작업을 수행
- 모듈화를 통해 시스템의 손쉬운 유지 관리 및 확장성 실현

ID	Interface	Struct_Name	Struct_comment	Field_cnt	Field_number	Filed_name	Field_type	Array_size	Array_byte	
									Start	Length
1										

ID	Interface	Struct_name	Struct_comment	Field_cnt	Field_number	Filed_name	Field_type	Array_size	Array_byte	
									Start	Length
2	UDP	SendToUnityDataPacket	Control S/W → Unity S/W	10	1	시퀀스레이션 설정	Int[]	8	0	12
					2	일부 지형 설정	Int[]	8	12	8
					3	객체 종류	Int[]	8	20	32
					4	객체 출발 X 좌표	Int[]	8	52	32
					5	객체 출발 Y 좌표	Int[]	8	64	32
					6	객체 목표 X 좌표	Int[]	8	116	32
					7	객체 목표 Y 좌표	Int[]	8	148	32
					8	객체 일부	Int[]	8	180	32
					9	객체 속성 #1	Int[]	8	212	32
					10	객체 속성 #2	Int[]	8	244	32

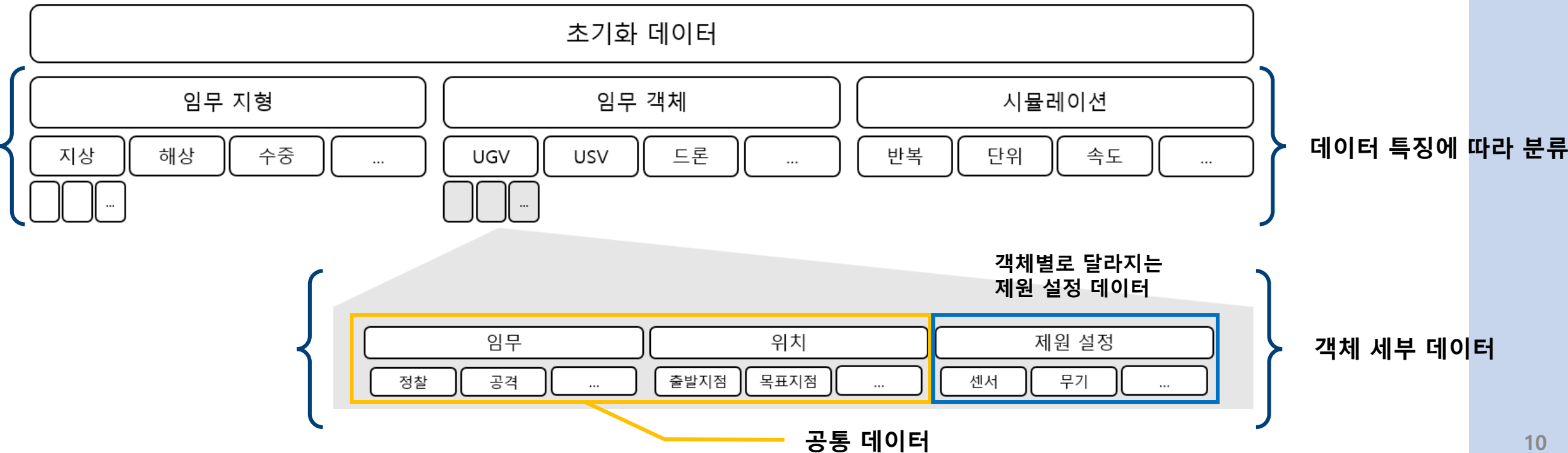
3. 연구 내용

❖ S/W 간 ICD 정의 – 초기화 데이터

ICD

초기화 데이터 구조

- ✓ 명령 SW에서 사용하는 초기화 데이터 정의
- ✓ 데이터 특징에 따라 크게 임무 지형, 임무 객체, 시뮬레이션으로 나눔
- ✓ 데이터 구조를 바탕으로 확장성을 고려하여 데이터 모듈화



3. 연구 내용

❖ S/W 간 ICD 정의 - Command S/W → Control S/W (초기화)

ID	Interface	Struct_name	Struct_comment	Field_cnt	Field_number	Filed_name	Field_type	Array_size	Array_byte	
									Start	Length
1	UDP	SendDataPacket	Command S/W → Control S/W	10	1	시뮬레이션 설정	Int[]	3	0	12
					2	임무 지형 설정	Int[]	2	12	8
					3	객체 종류	Int[]	8	20	32
					4	객체 출발 X 좌표	Int[]	8	52	32
					5	객체 출발 Y 좌표	Int[]	8	84	32
			공통 데이터		6	객체 목표 X좌표	Int[]	8	116	32
					7	객체 목표 Y좌표	Int[]	8	148	32
					8	객체 임무	Int[]	8	180	32
			객체별 제원 설정 데이터		9	객체 속성 #1	Int[]	8	212	32
					10	객체 속성 #2	Int[]	8	244	32

✓ 필요에 따라 객체 속성을 n개로 증가시킬 수 있음

3. 연구 내용

❖ S/W 간 ICD 정의 - Control S/W → Simulation S/W(초기화)

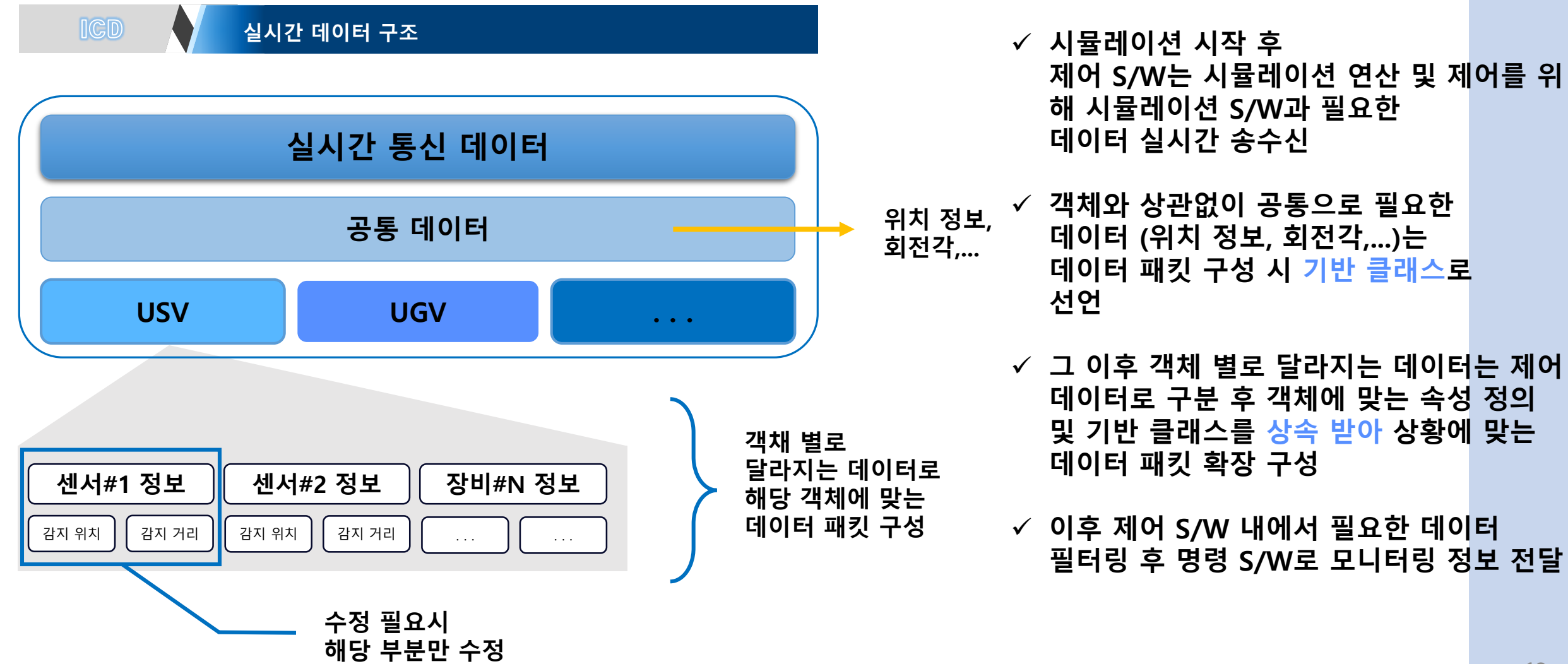
ID	Interface	Struct_name	Struct_comment	Field_cnt	Field_number	Filed_name	Field_type	Array_size	Array_byte	
									Start	Length
2	UDP	SendToUnity DataPacket	Control S/W → Simulation S/W	10	1	시뮬레이션 설정	Int[]	3	0	12
					2	임무 지형 설정	Int[]	2	12	8
					3	객체 종류	Int[]	8	20	32
					4	객체 출발 X 좌표	Int[]	8	52	32
					5	객체 출발 Y 좌표	Int[]	8	84	32
					6	객체 목표 X좌표	Int[]	8	116	32
					7	객체 목표 Y좌표	Int[]	8	148	32
					8	객체 임무	Int[]	8	180	32
					9	객체 속성 #1	Int[]	8	212	32
					10	객체 속성 #2	Int[]	8	244	32

0: UGV
1: USV
2: 드론
⋮

✓ 객체 종류 인덱스를 사용해서
개체 종류 파악
✓ 객체 파악 후 객체에 맞는
속성 정의

3. 연구 내용

❖ S/W 간 ICD 정의 - 실시간 통신 데이터



3. 연구 내용

❖ S/W 간 ICD 정의 - Simulation S/W → Control S/W (실시간) 예시 (USV)

ID	Interface	Struct_name	Struct_comment	Field_cnt	Field_number	Filed_name	Field_type	Array_size	Array_byte	
									Start	Length
3	UDP	USVDataPacket ToControl	Simulation S/W → Control S/W	11	1	현재 위치 x	Float	1	0	4
				}	2	현재 위치 y	Float	1	4	4
					3	현재 위치 z	Float	1	8	4
					4	Euler 각 x	Float	1	12	4
					5	Euler 각 y	Float	1	16	4
					6	Euler 각 z	Float	1	20	4
					7	감지된 장애물 수	Int	1	24	4
				}	8	감지된 장애물의 x 좌표	Float[]	10	28	40
					9	감지된 장애물의 y 좌표	Float[]	10	68	40
					10	감지된 장애물의 z좌표	Float[]	10	108	40
					11	감지된 장애물과의 거리	Float[]	10	148	40

공통 데이터

객체별 맞춤
데이터

3. 연구 내용

❖ S/W 간 ICD 정의 – Control S/W → Simulation S/W (실시간) 예시 (USV)

ID	Interface	Struct_name	Struct_comment	Field_cnt	Field_number	Filed_name	Field_type	Array_size	Array_byte	
									Start	Length
4	UDP	USVDataPacket ToUnity	Control S/W → Simulation S/W	4	1	목표 위치 x	Float	8	0	32
					2	목표 위치 y	Float	8	32	32
					3	제어 데이터 #1	Float	8	64	32
					4	제어 데이터 #2	Float	8	96	32

공통 데이터

객체별 맞춤
데이터

✓ 필요에 따라 제어 데이터를 n개로 증가시킬 수 있음

3. 연구 내용

❖ S/W 간 ICD 정의 – Control S/W 내부 통신 (실시간) 예시

ID	Interface	Struct_name	Struct_comment	Field_cnt	Field_number	Filed_name	Field_type	Array_size	Array_byte	
									Start	Length
5		MessageToModel	SimModelComponent → Simulation Model	2	1	현재 위치 x	int	1	0	4
					2	현재 위치 z	int	1	4	4

✓ 컴포넌트에서 모델에게 필요한 모니터링 정보를 전달하면 I/O 인터페이스를 통해 Command S/W로 전달

❖ S/W 간 ICD 정의 – Control S/W -> Command S/W 통신 (실시간) 예시

ID	Interface	Struct_name	Struct_comment	Field_cnt	Field_number	Filed_name	Field_type	Array_size	Array_byte	
									Start	Length
6	UDP	SendToWF Datapacket	Control S/W → Command S/W	2	1	현재 위치 x	int[]	8	0	32
					2	현재 위치 z	int[]	8	32	32

3. 연구 내용

❖ 명령 s/w 레이아웃 및 기능

1

임무 설정

2

객체 속성 설정

3

시뮬레이션 속성 설정

4

CLOSE

5

POSITION
ASSIGNMENT

6

REAL TIME
INFORMATION

초기 위치 설정

위치 정보 할당

실시간 위치 출력

USV_1

USV_2

USV_3

3

COORDINATE

START X

259

START Y

353

FINAL X

517

FINAL Y

468

OBJECT MISSION

DISASTER RESPONSE

COMMUNICATION

OFF

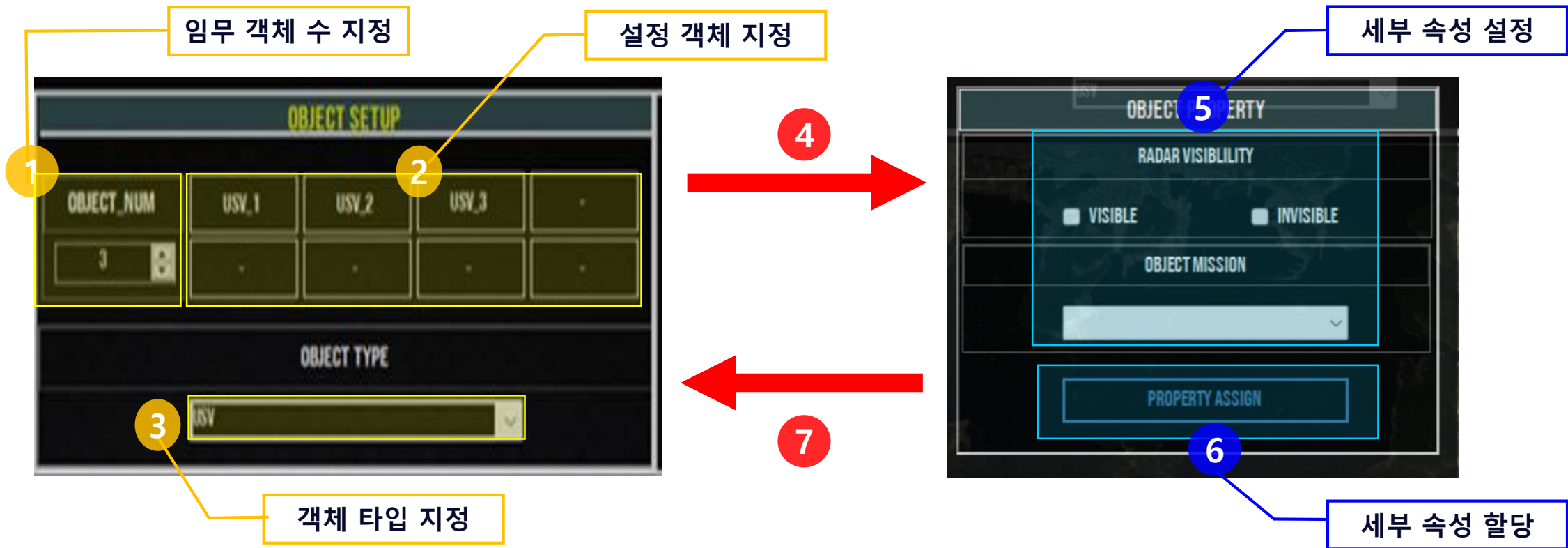
종료

EXIT

설정 객체 지정

3. 연구 내용

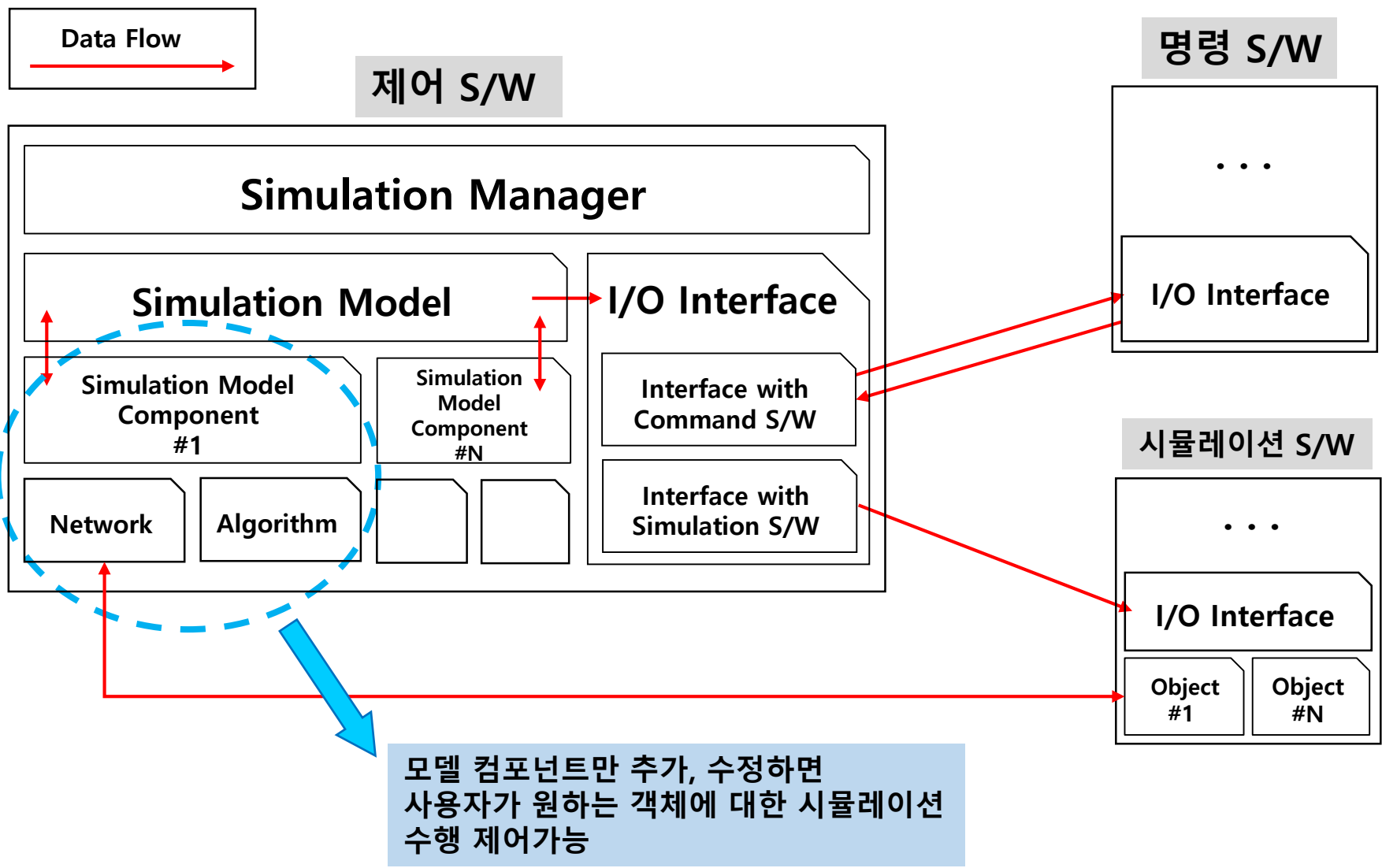
❖ 명령 s/w 레이아웃 및 기능



- ✓ 객체 설정 후 설정 객체에 맞는 세부 속성 설정 팝업 업로드
- ✓ 객체마다 설정하고 싶은 속성 자유롭게 추가 및 수정 가능

3. 연구 내용

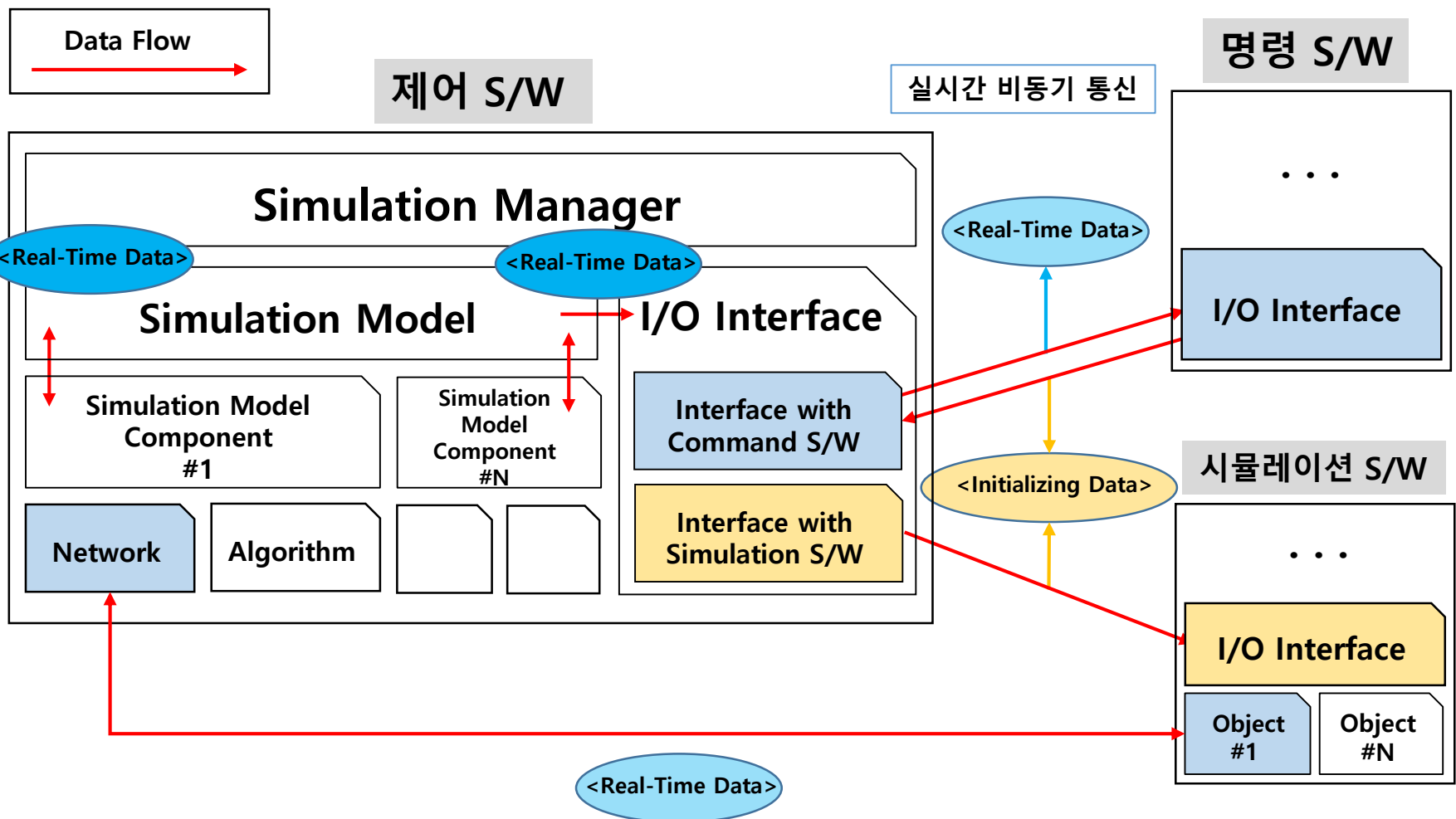
❖ 제어 S/W 구조 및 설명



- ✓ 제어 S/W 가 시뮬레이션을 구동하는 핵심 역할이기 때문에 시뮬레이션에 필요한 기능적 요소들을 담아 구조 설계
- ✓ 시뮬레이션 매니저는 제어 S/W 최상단에 위치하여 시뮬레이션 초기화 작업 및 인터페이스&모델 등록/제어, 시뮬레이션 시작 및 종료 등의 역할 담당
- ✓ 시뮬레이션 모델은 사용자가 모의하기 원하는 대상으로, 여러가지 시뮬레이션 모델 컴포넌트의 조합으로 이루어져 있음. 다양한 컴포넌트 동작의 조합을 통해 시뮬레이션을 진행
- ✓ 추후 사용자는 시뮬레이션 모델 컴포넌트만 구현하면 새로운 객체에 대한 시뮬레이션을 원활히 수행할 수 있음

3. 연구 내용

❖ 제어 S/W 구조 - 데이터 흐름

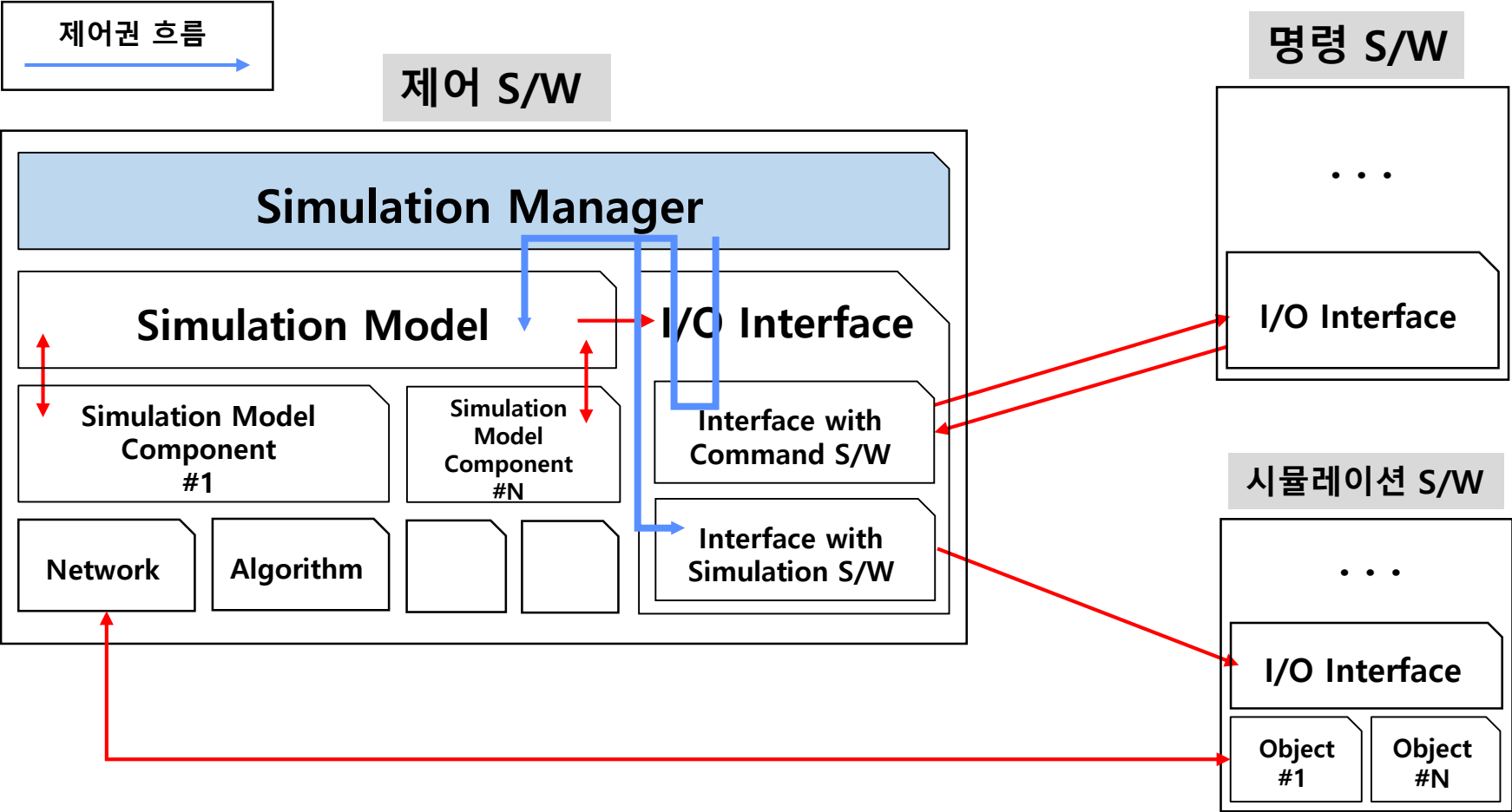


- ✓ 제어 S/W의 I/O 인터페이스는 명령 S/W, 그리고 시뮬레이션과의 통신을 담당
- ✓ 시뮬레이션 시작 후, 시뮬레이션 모델 컴포넌트와 그에 대응하는 시뮬레이션 S/W 내부의 객체는 1:1로 대응하여 통신
- ✓ 모델 컴포넌트의 종류가 다양하기 때문에, 사용하는 데이터 패킷의 종류가 달라 이를 한번에 모아서 처리하고 수신 측에서 다시 자신의 데이터를 찾는 과정을 생략하고, 1:1로 직접 통신

3. 연구 내용

❖ 제어 S/W 실행 순서

➢ 시뮬레이션 초기화 및 실행 단계



- ✓ 시뮬레이션 매니저를 중심으로 하는 초기화 단계 흐름
- ✓ 매니저가 인터페이스를 등록한 후 명령 S/W로부터 초기화 데이터를 수신
- ✓ 수신 데이터를 기반으로 모델 초기화 및 시뮬레이션 S/W로 초기 데이터 전송
- ✓ 이후 시뮬레이션이 시작되면 각 컴포넌트의 네트워크 및 알고리즘 수행
- ✓ 모델은 명령 S/W와의 인터페이스를 참조해 필요한 데이터 전송

3. 연구 내용

❖ 제어 S/W 내부 시뮬레이션 모델과 컴포넌트의 유연성

➤ 시뮬레이션 모델 내부 멤버

```
class SimModel
{
    static public List<ISimComponent> compoList = new List<ISimComponent>();
    static public List<MessageToModel> messageList = new List<MessageToModel>();

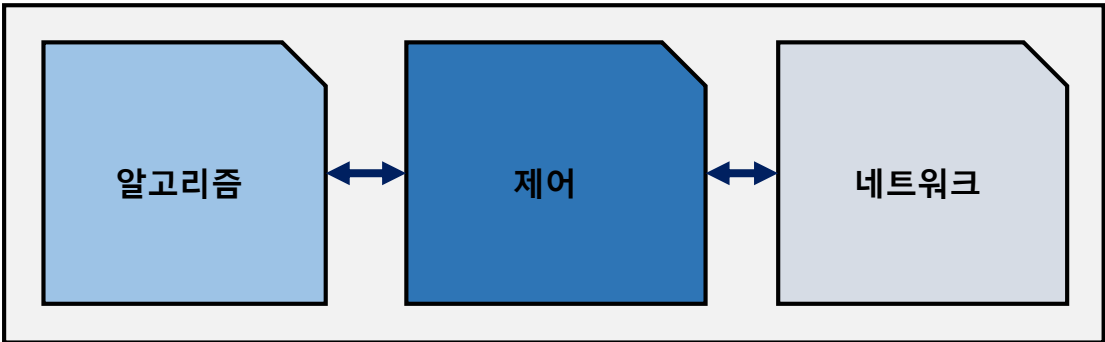
    public SimInterfaceWithCommand simInterfaceWithCommand;
    public SendToWFDatapacket sendToWFDatapacket;
}
```

여러 컴포넌트의 개수를 동적으로 변경하며 제어

명령 S/W 인터페이스 등록 후, 필요한 데이터만 Marshalling 후 전송

시뮬레이션 모델과 컴포넌트의 독립성 확보

➤ 시뮬레이션 모델 컴포넌트 구조



- ✓ 시뮬레이션 모델 컴포넌트는 총 3가지 파트 (제어, 알고리즘, 네트워크)로 구성
- ✓ 제어 파트를 통해 컴포넌트 알고리즘 수행 및 모델과의 통신 통제
- ✓ 알고리즘 파트로 다양한 내부 로직 구현
- ✓ 네트워크 파트는 시뮬레이션 S/W와의 1:1 통신
- ✓ 각자 독립성을 가지며 구현, 추후 다양한 Logic 추가 및 수정 시 알고리즘 파트만 수정

3. 연구 내용

❖ 시뮬레이션 모델과 컴포넌트의 유연성

➤ 시뮬레이션 모델 컴포넌트 내부 멤버

```
class USVDemo : ISimComponent
{
    //내부적으로 필요한 데이터 및 전송 데이터
    public int startX, startZ; // 시작 위치
    public int dx, dz; // 목표 지점

    //디텍트 리스트
    List<DetectedInfo> detectList = new List<DetectedInfo>();

    //해검 앞뒤 양 옆 이동 제어하는 데이터
    public float v; //선속도
    public float w; //각속도

    public Velocity _velocity;

    //수신데이터
    public float cx, cy, cz; //수신 데이터 (해검 현재 위치)
    public float ex, ey, ez; //오일러 각도
    public int detectedNumber;
    public float[] o1, o2, o3;
    public int[] life;
    public float[] distance;

    //모델과 통신하기 위한
    public MessageToModel messageToModel;

    //유니티와 1대1 통신하기 위한 용도
    public Network myNetwork;
    public SendToUnityDataPacket myPacket; //보내는거
    public SendFromUnityDataPacket oppPacket = new SendFromUnityDataPacket(); //받는거

    //DWA
    DWAdemo dwa = new DWAdemo();
}
```

ISimComponent 인터페이스
상속을 통한 모델의 개별 컴포넌트
접근 허용

컴포넌트 내부적으로 필요한
데이터와 시뮬레이션 모델과의
통신 메시지 정의

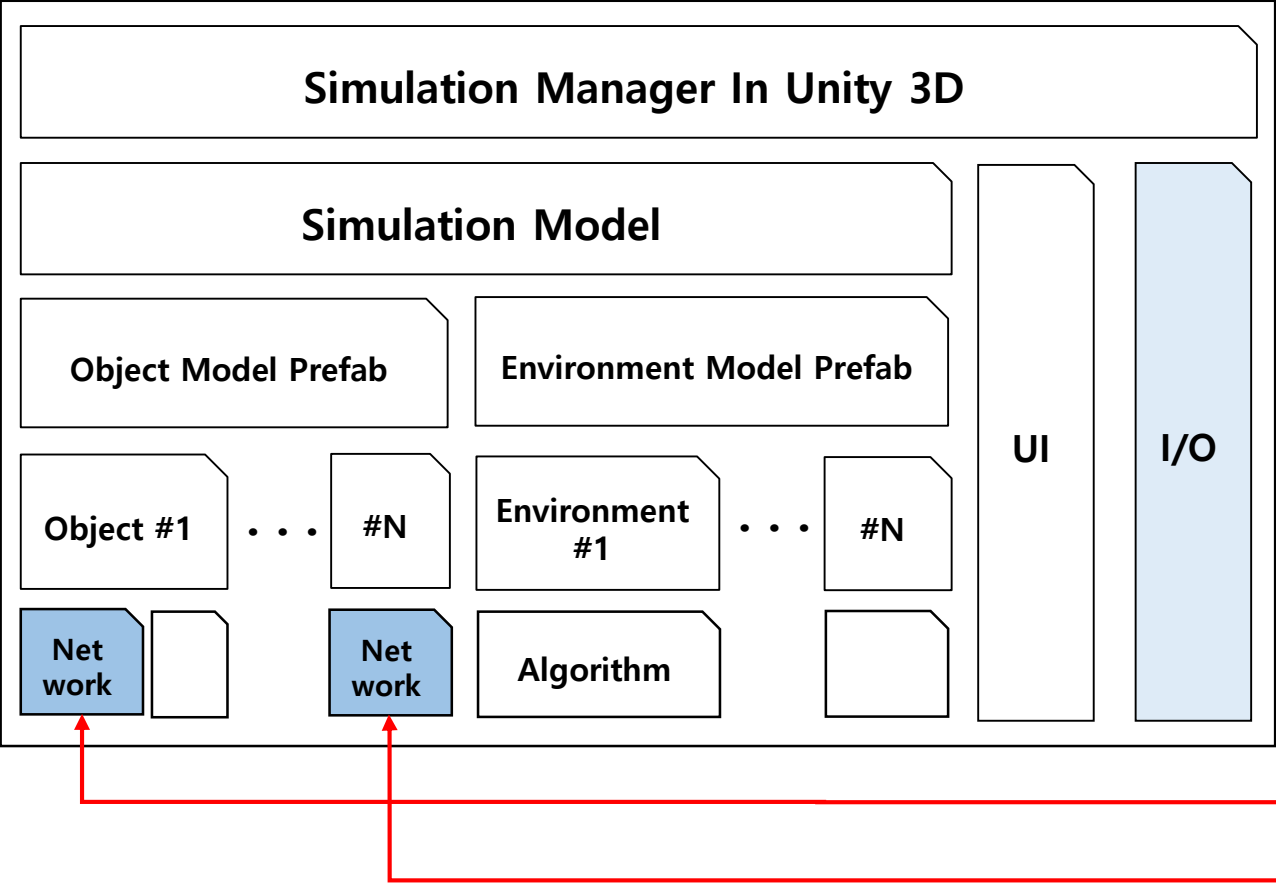


유니티와의 통신을 위한
네트워크 모듈 및 데이터 패킷

3. 연구 내용

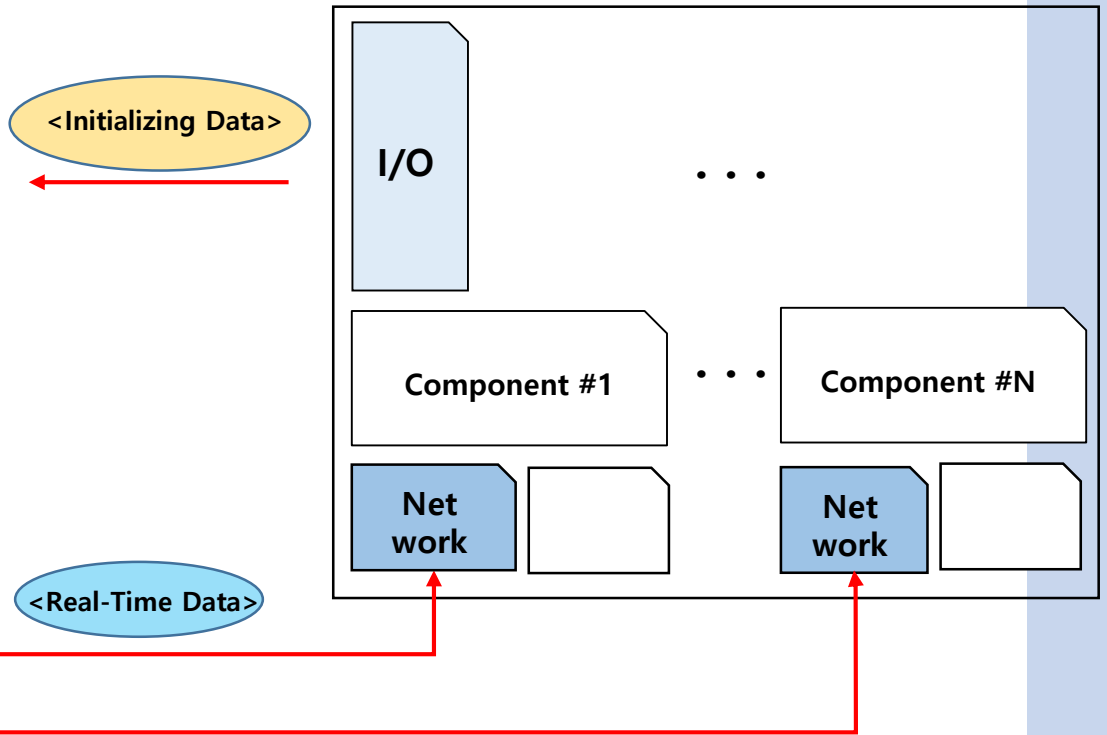
❖ 시뮬레이션 S/W 구조 (Unity 3D)

시뮬레이션 S/W



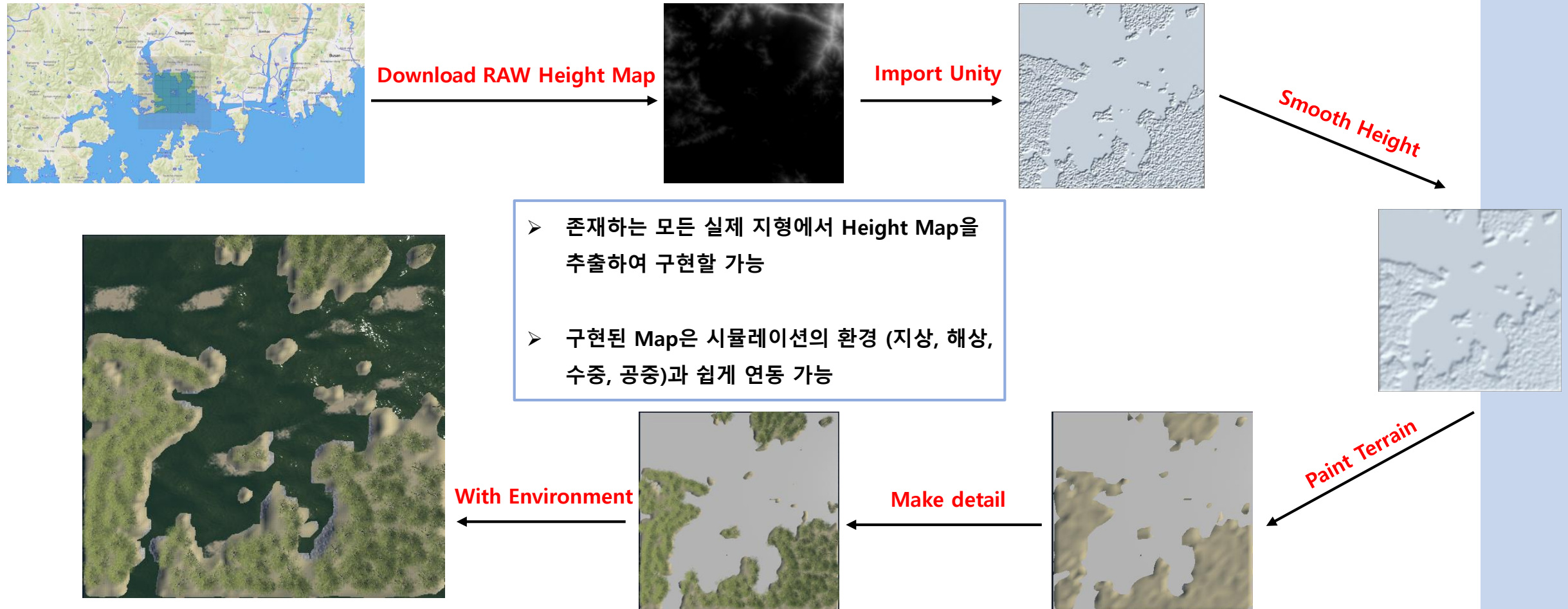
- Unity 내부 시뮬레이션 매니저가 시뮬레이션 S/W 제어
- 초기화 데이터 수신 후 Prefab을 활용해 객체 동적 생성
- 시뮬레이션 시작 후, 각 컴포넌트들은 제어 S/W와 1대1 대응 통신

제어 S/W



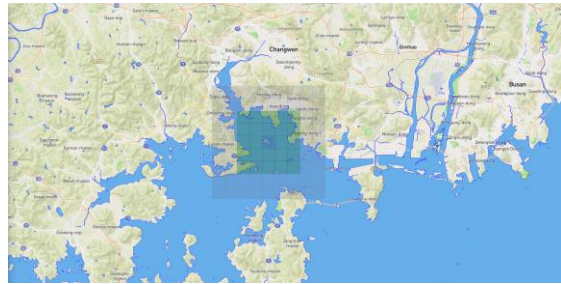
3. 연구 내용

❖ 해상 환경 속 지형 구현 - Use RAW Height Map



3. 연구 내용

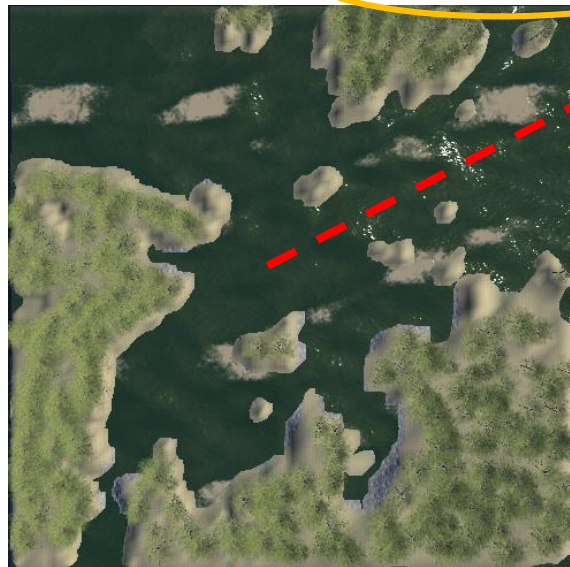
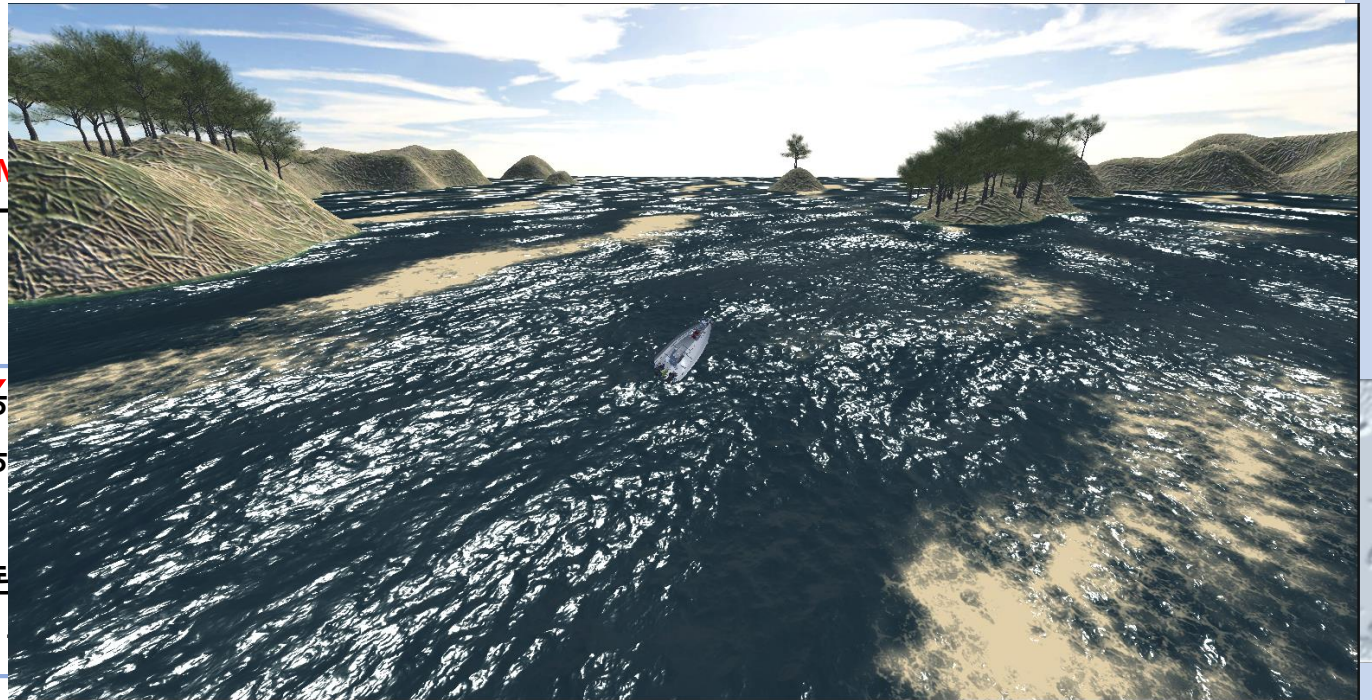
❖ 해상 환경 속 지형 구현 - Use RAW Height Map



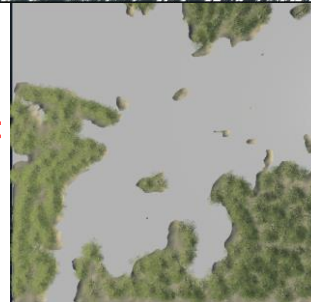
Download RAW Height Map

In Simulation

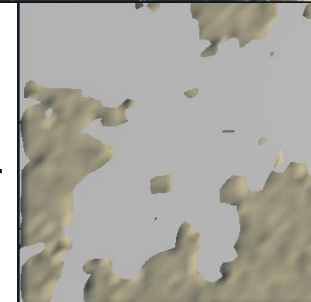
- 존재하는 지형 데이터를 추출하여
- 구현된 수중, 육상 지형에 적용



With Environment



Make detail



Paint Terrain

4. 연구 산출물

4. 연구 산출물

❖ 연구 산출물 현황

» 논문 1편

- - 제목: GCS interface mission load reduction design through pilot gaze point analysis
- 저자: 이세담, 허진영, 김주영, 권용진
- 학술지명: International Journal of Mechanical Engineering and Robotics Research (IJMERR)
- 2022년도 하반기 게재 예정

» 특허 출원

- - 제목: 무인항공기 원격운용자의 비정상 상태에 따른 실수 및 오작동 방지 알고리즘
- 발명자: 이세담, 권용진, 유재관, 허진영, 이경하
- 2022년도 출원 예정

» 보고서 및 S/W Code 압축 파일

- 5.31 까지 제출 예정

5. 향후 연구 계획

5. 향후 연구 계획

❖향후 연구 계획 (~22.10)

» AI 학습 연동을 위한 AI 모듈 설계 및 구현

- ✓ 딥러닝, 강화 학습 등 AI 알고리즘 학습을 제어 S/W와 연동할 수 있는 모듈 구현

» 다양한 환경 및 플랫폼 추가 모델링 및 구현

- ✓ 수중, 지상 등 다양한 환경에서의 시뮬레이션을 위한 환경 구축
- ✓ 수중 로봇, 드론, UGV 등 다양한 플랫폼 모델링 및 구현

» 원격 통신을 통한 시뮬레이션 수행

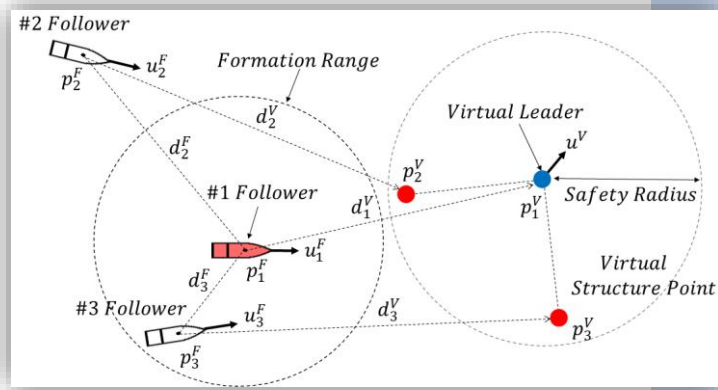
- ✓ 원격 통신 최적화 작업을 통해 서로 다른 컴퓨터에서 모듈화 된 S/W를 사용해 원격 시뮬레이션 수행

» 이기종 플랫폼 간 군집 제어 알고리즘 연구 및 시나리오 검증

- ✓ 모듈화 된 S/W 에서 이기종 군집 제어 시나리오 수행을 통한 시뮬레이터 검증 수행

◆ 수중 환경 예시

◆ 군집 제어 알고리즘 예시



감사합니다