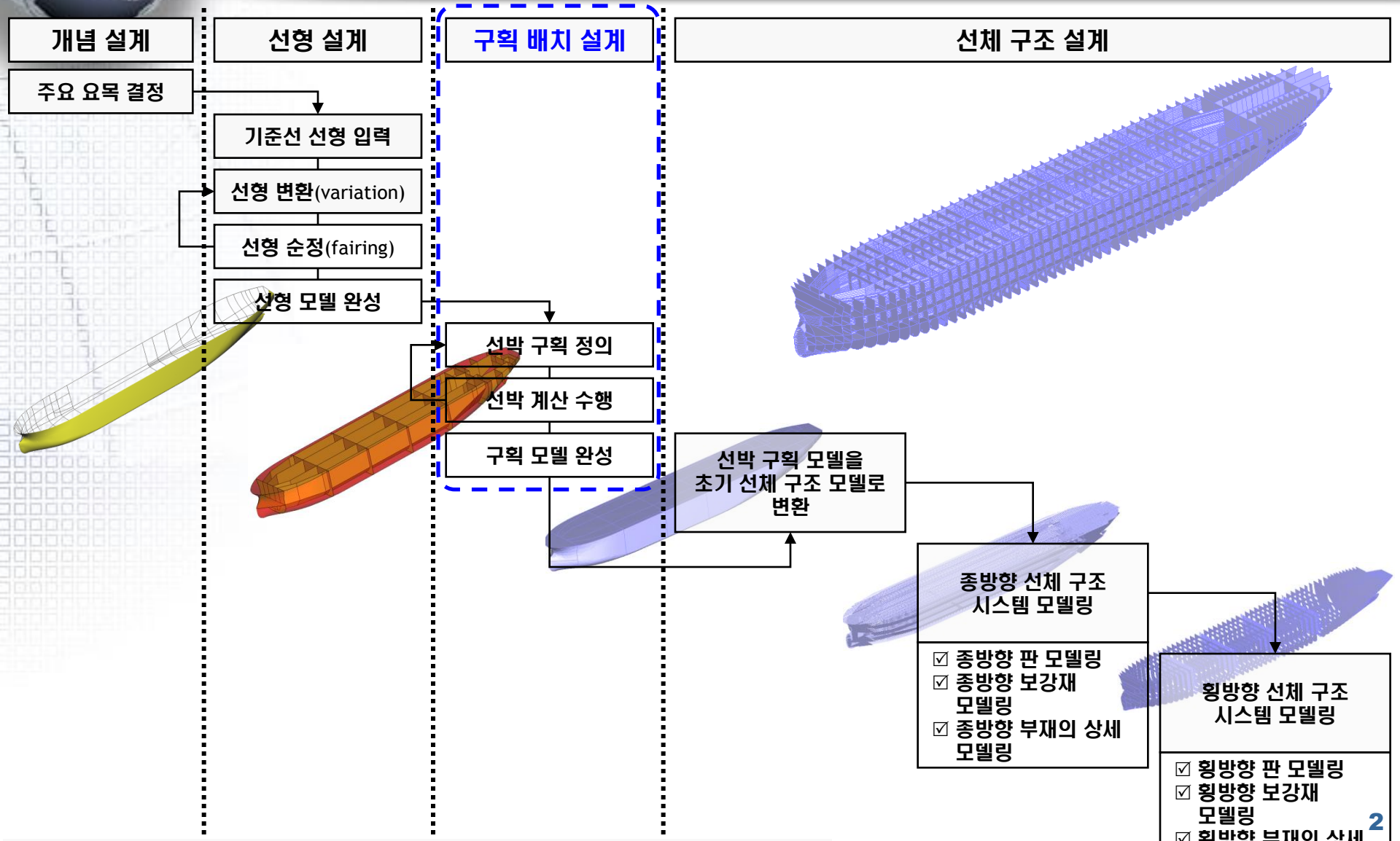
General Arrangement Design using “EzCompart CAD program
(구획 배치 설계 프로그램 “EzCOMPART”를 이용한 배치설계)

2008. 5.

서울대학교 조선해양공학과
이규열

NAOE/SNU
@SDAL Advanced Ship Design Automation Lab.
http://asdal.snu.ac.kr

선박 제품 설계(선형/구획/구조) 절차



* 종방향 선체 구조 시스템: 외판 시스템, 갑판 시스템, 거더 시스템, 스트링거 시스템, 종격벽 시스템 등
* 횡방향 선체 구조 시스템: 횡격벽 시스템, 웨브 프레임 시스템

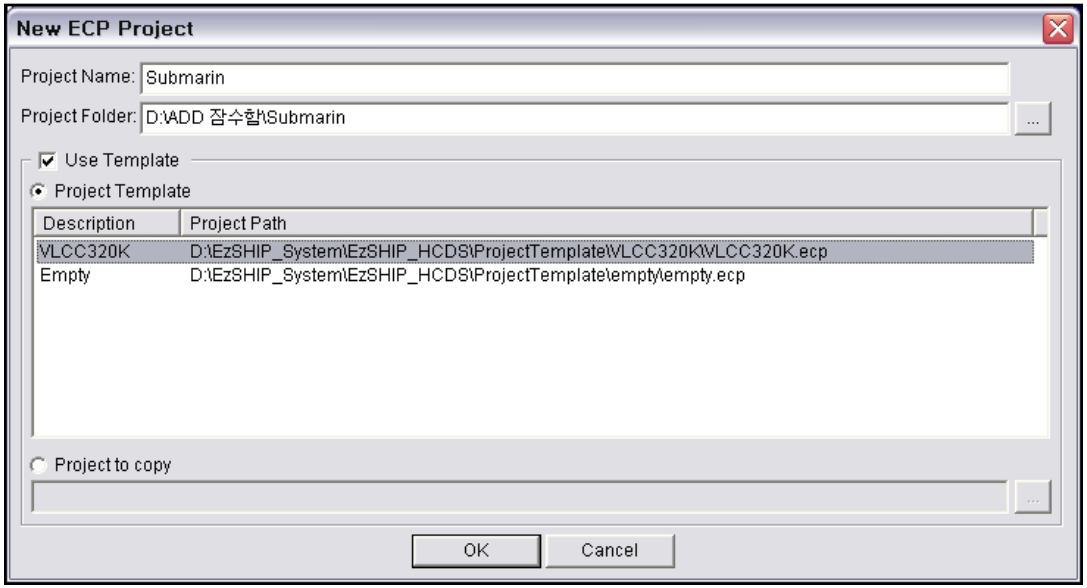
배치 설계 시스템의 화면 구성(1)

배치 설계 시스템의 실행

- File **메뉴**
  →Create EzCOMPART Workspace

새로운 Project **만들기**

- File **메뉴** → New project
- **아이콘**
- **기존의** template **복사**



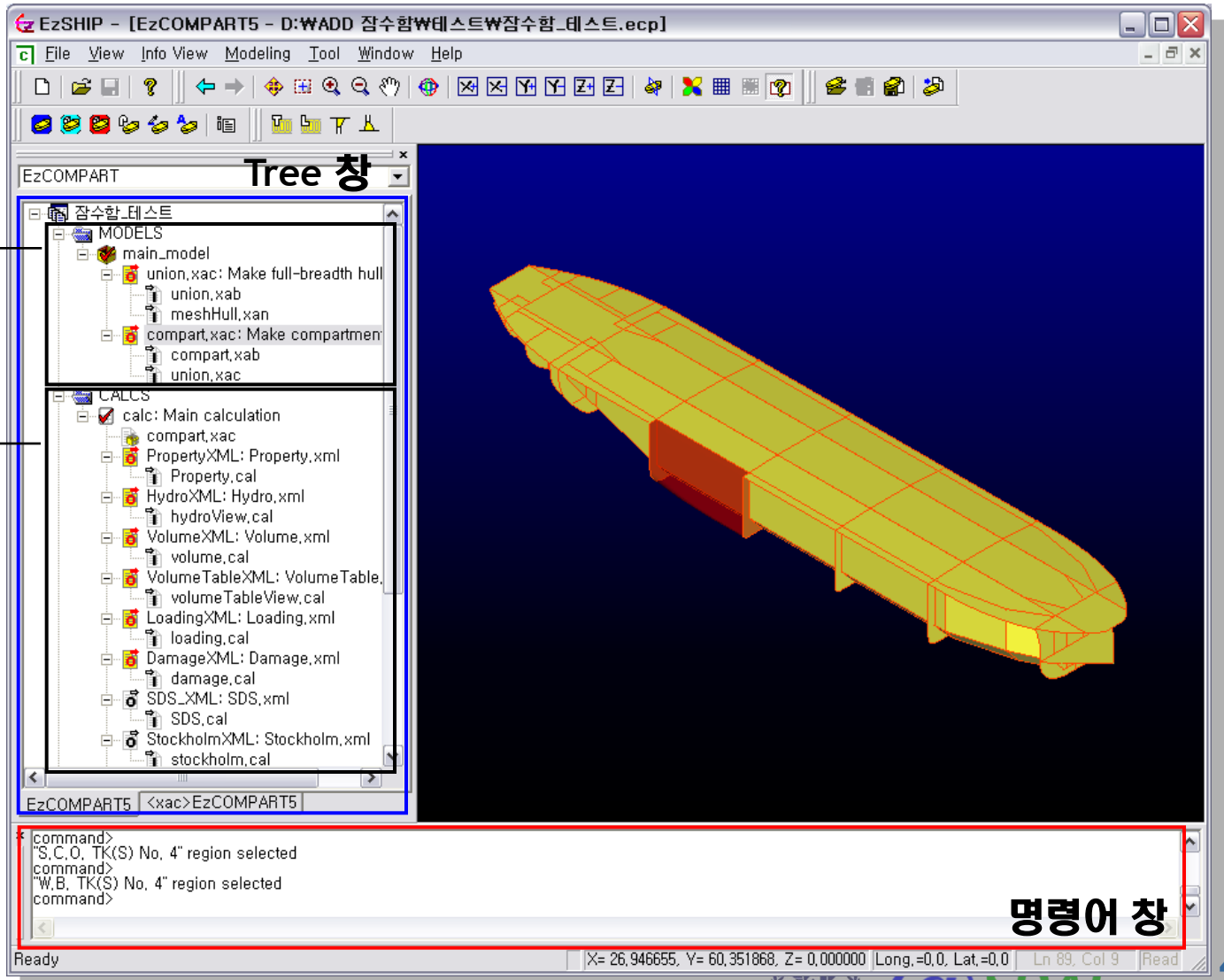
Project **열기**

- File **메뉴** → Open project
- **아이콘**
- *.ecp **파일**

배치 설계 시스템의 화면 구성(2)

선박 구획 모델과 관련된 부분

선박 계산과 관련된 부분



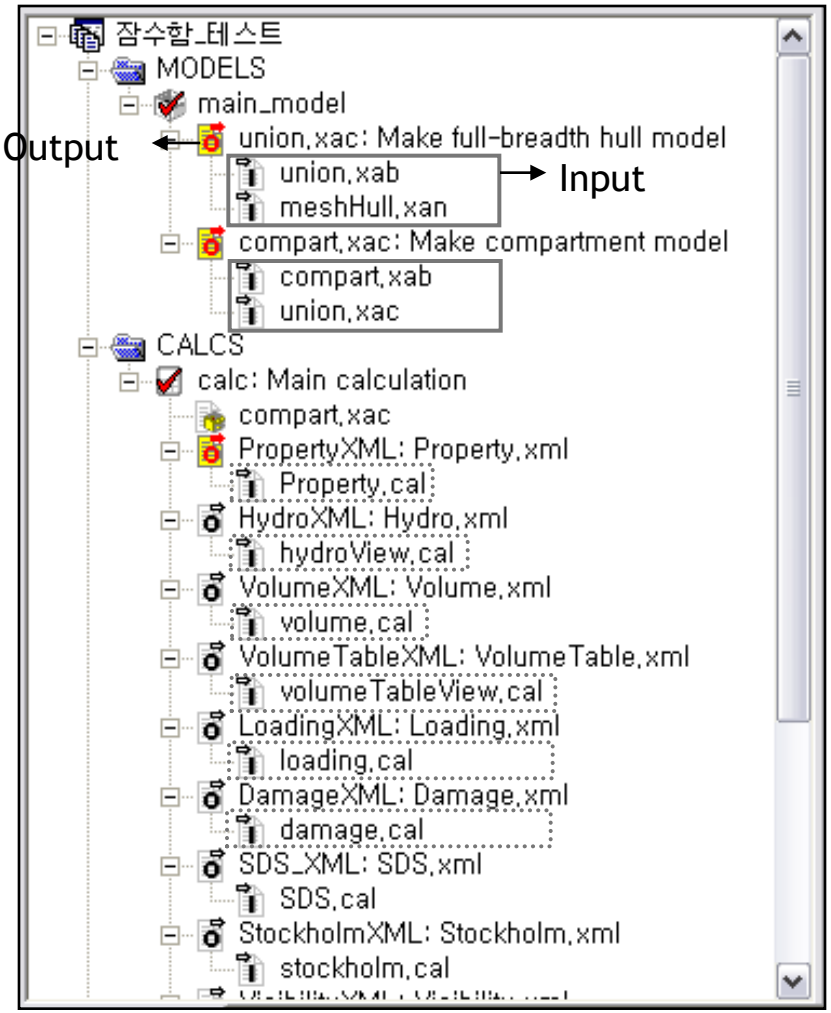
배치 설계 시스템의 화면 구성(3)

선박 구획 모델의 정의

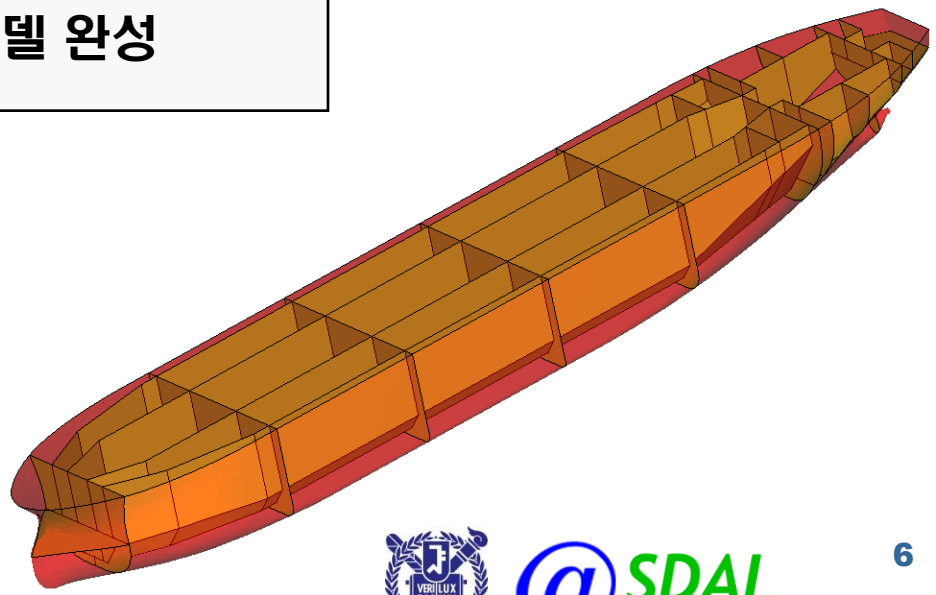
- EzHULL **반폭 메쉬 선형 파일 이용**
- **전폭으로 확장, 선형 확보**(union.xab)
- **구획 정의 스크립트 작성** (compart.xab)
- **구획 정의 모델 확보**

선박 계산

- **일반 상선**
  Hydro Static, Volume, Loading, Damage **계산**
- **컨테이너선**: SDS **계산**
- **로로**(RO-RO)**선 등**: Stockholm **계산**
- *.cal **파일 작성** → xml **형식의** Output

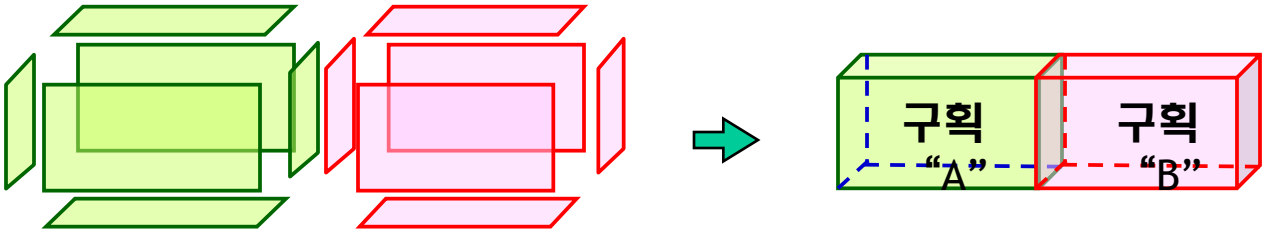


배치 설계 과정

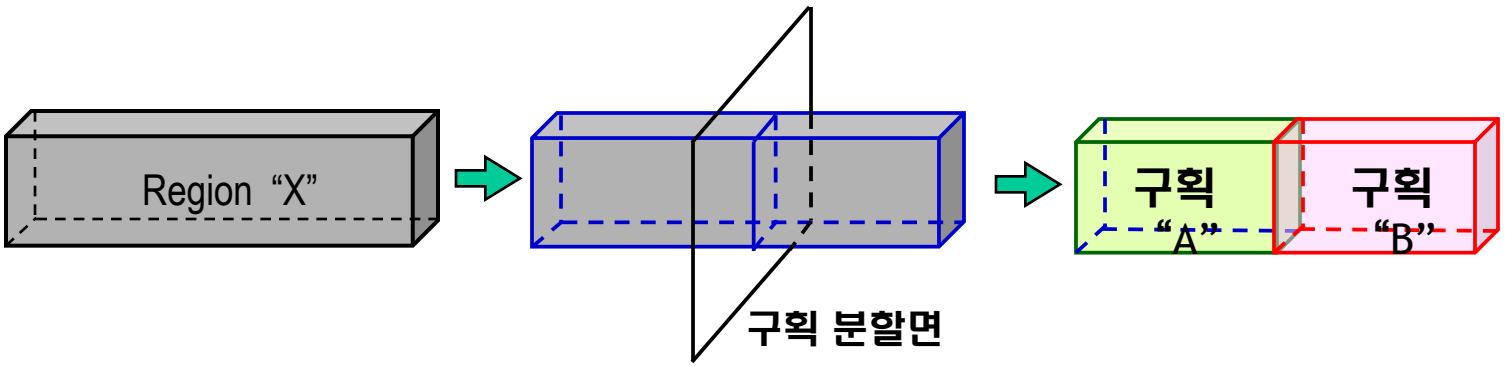


1. 선박 구획 정의
- 구획 정의의 기본 개념

기존 시스템: 개별 구획 중심 정의



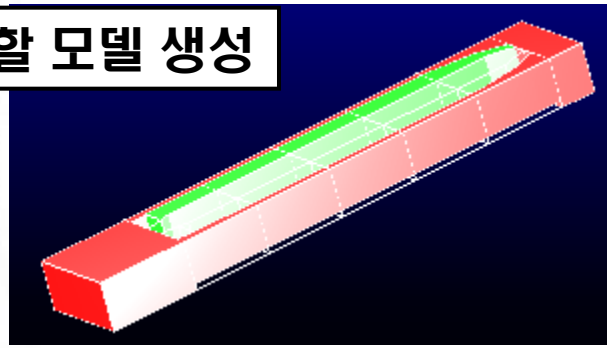
본 시스템: 구획 분할면 중심 정의



1. 선박 구획 정의
- 선박 구획 모델링 절차

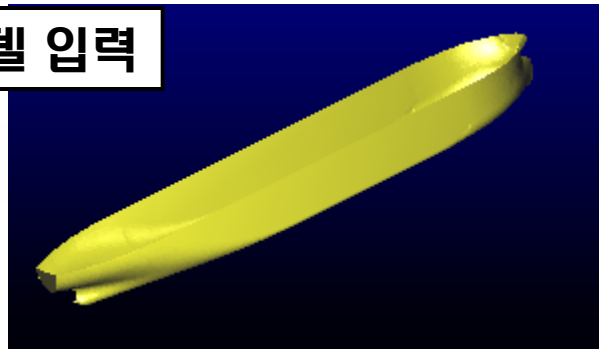
선박 구획 모델 생성 과정

1. 구획 분할 모델 생성



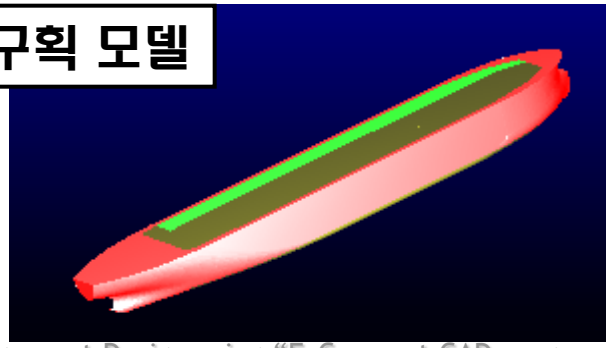
+

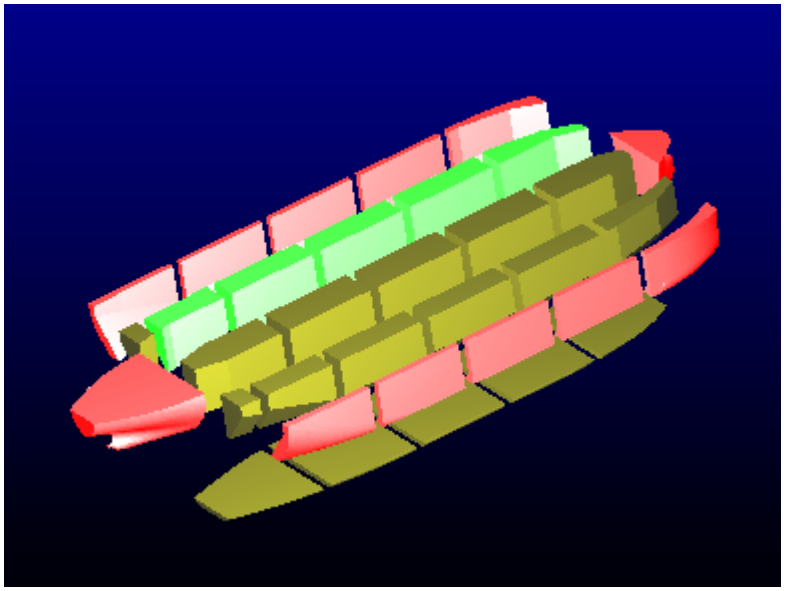
2. 선형 모델 입력



⇩

3. 완성된 구획 모델



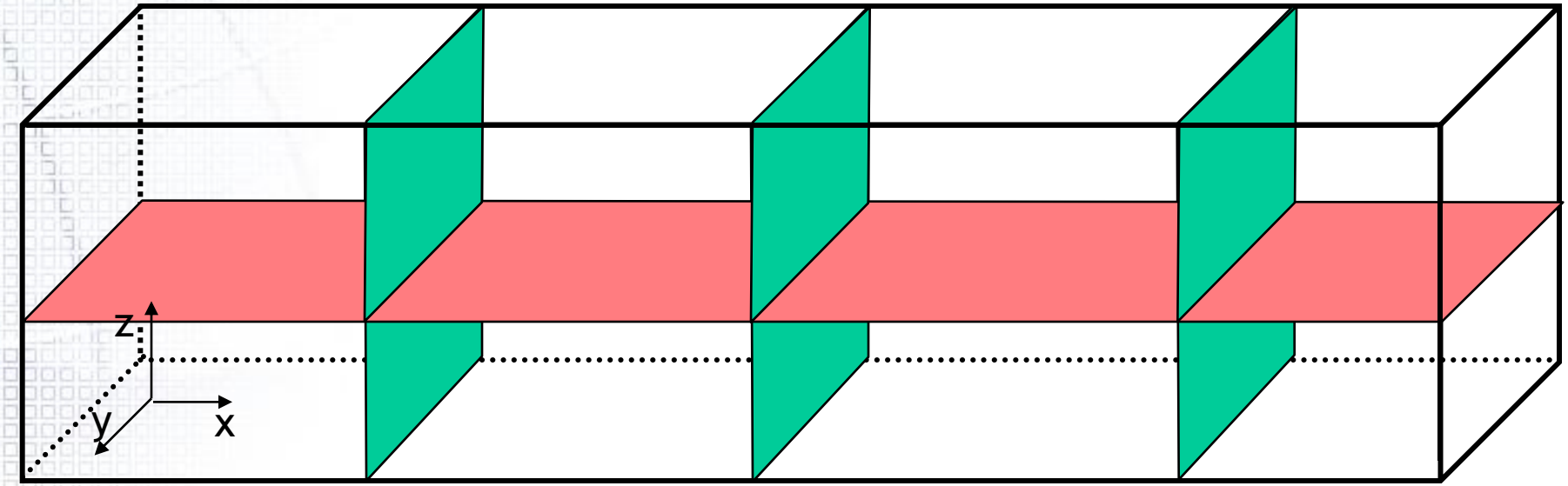


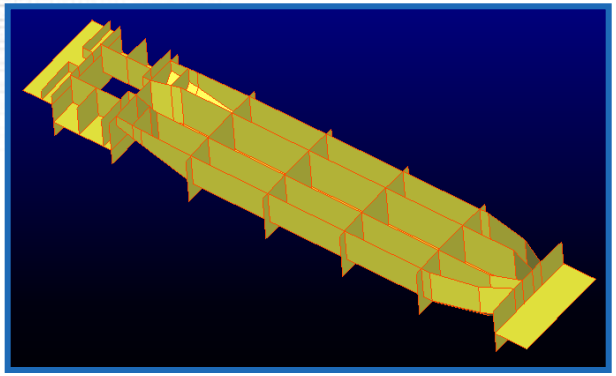
완성된 구획 모델을 전개한 모습

1. 선박 구획 정의
- 구획 모델링의 개념(1)

선박보다 약간 더 큰 Box

X plane, Y plane, Z plane **과 같은 격벽**(cutter) **정의 → 구획분할**

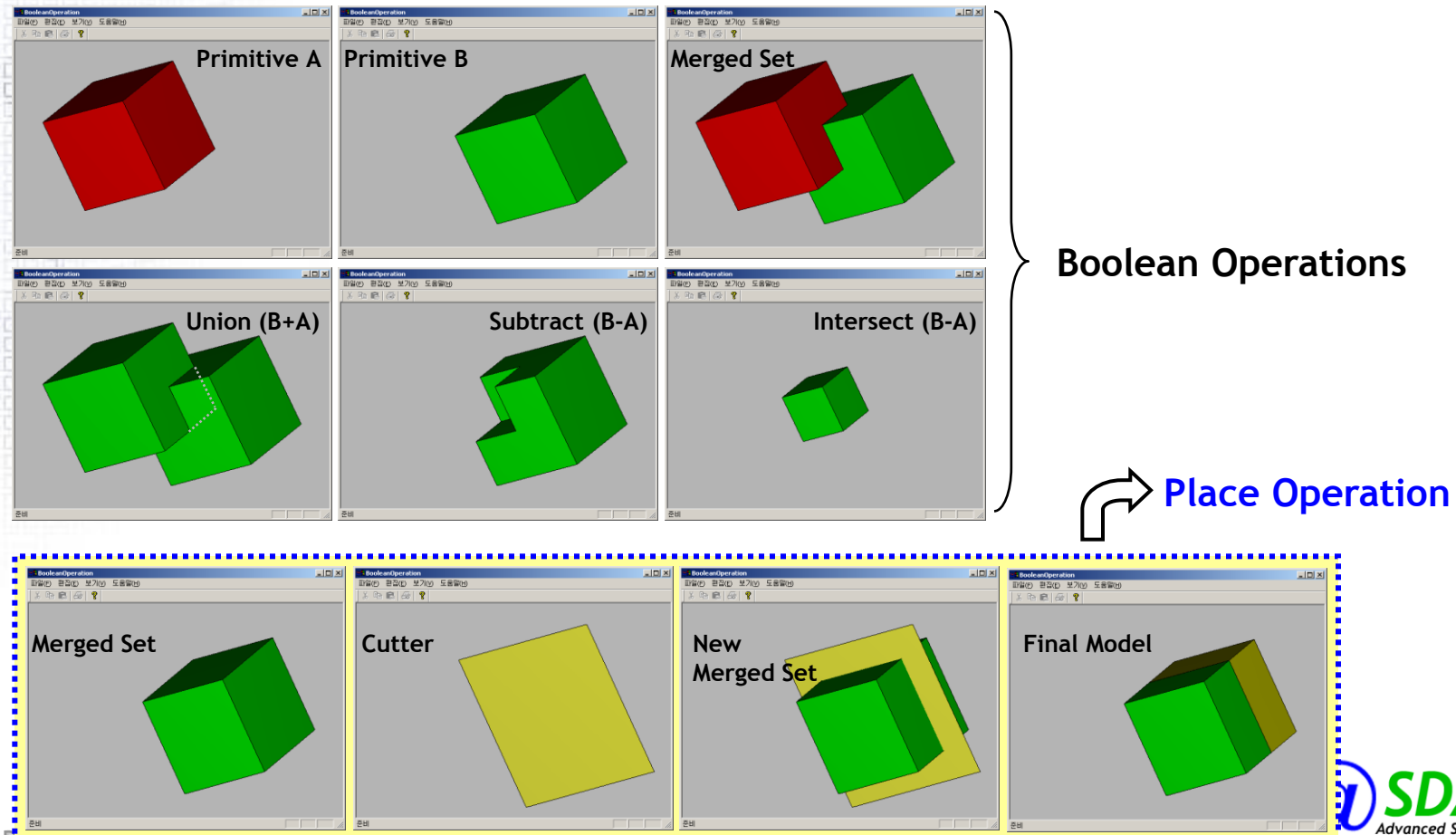




구획을 정의하는 과정 =
구획을 나누는 격벽 (cutter)**를 정의하는 과정**

1. 선박 구획 정의
- 구획 모델링의 개념(2)

앞서 정의된 구획 분할면을 이용하여 기존 구획을 분할할 수 있는 확장된 구획 정의 연산인 place를 제공하여 대부분 구획을 생성하고, 솔리드 모델간의 기본적인 Boolean연산(합집합-union, 교집합-intersect, 차집합-subtract)도 제공한다.



배치 설계 시스템의 구획 정의 파일 - “base” 블록: 연산에 의한 구획 정의(2)

명령어 형식

union 대상 모델 이름; // base로 설정된 대상 모델에 대해 합집합 연산 수행
subtract 대상 모델 이름; // base로 설정된 대상 모델에 대해 차집합 연산 수행
intersect 대상 모델 이름; // base로 설정된 대상 모델에 대해 교집합 연산 수행

사용 예

```
base hull {
    box container_box { (0,  0, 0), (6, 2, 2) };  // 길이 =6, 높이=폭=2인 Box 정의
    union container_box;          // 앞에서 정의된 Box를 base 모델인 hull과 합집합 연산을
    ....                          // 수행한 모델을 생성
}
```

✓ 주의 사항 : 배치 설계 시스템의 복합 다양체모델 커널의 특성상, Boolean 연산의 대상은 Box 뿐만 아니라 xplane(y,z), reflectsurf, skinsurf, polyline으로 만들어지는 모델도 해당된다.

사용 예

```
base hull {
    reflectsurf rhull y=0 hull;
    union rhull;
}
```

1. 선박 구획 정의
- 구획 모델링의 개념(3)

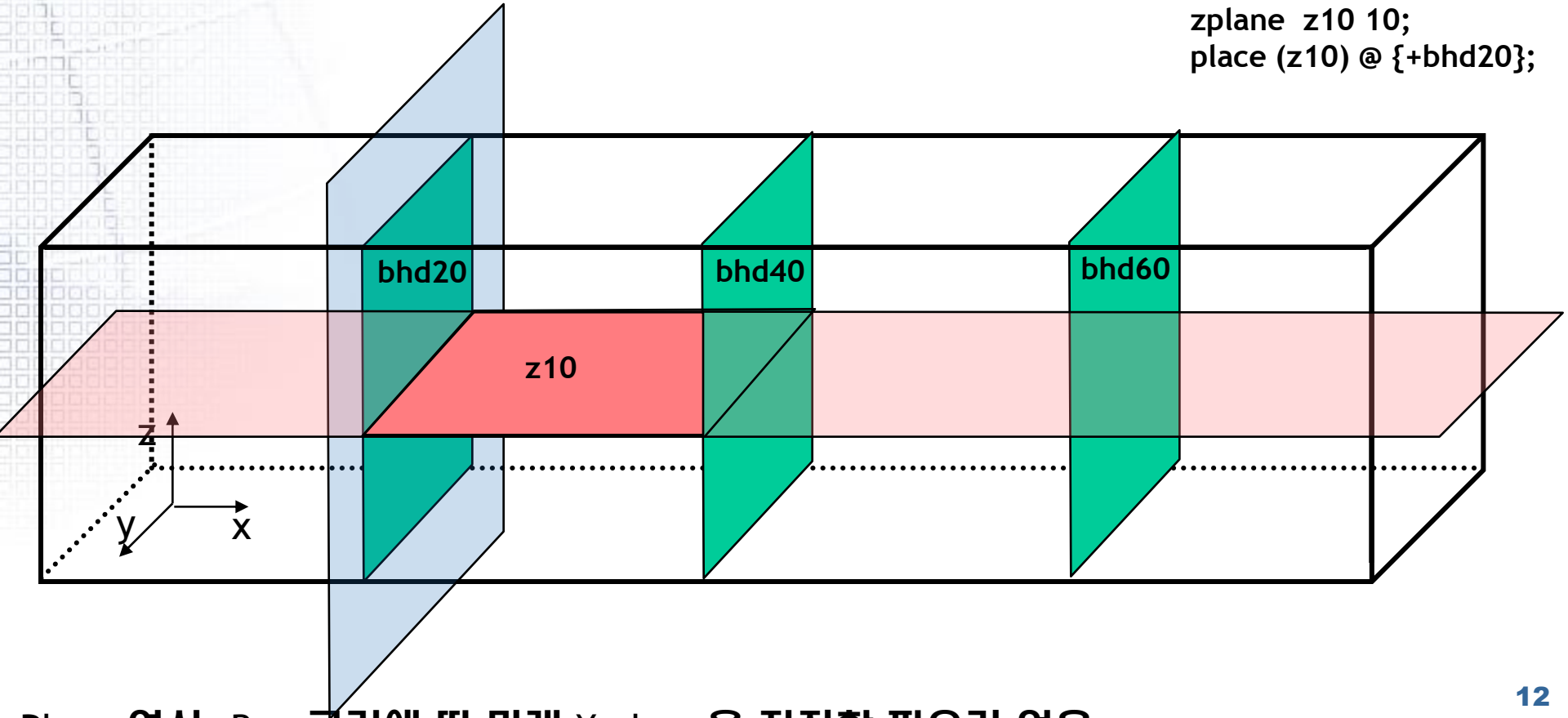
Place 연산의 예

(Box 안의 구획) Place (X plane 무한 평면) = (파란색 격벽)

```
xplane  bhd20 20;
place (bhd20) @ {+box};
```

(bhd20~bhd40 격벽 사이 구획) Place (Z plane 무한 평면) = (빨간색 격벽)

```
zplane  z10 10;
place (z10) @ {+bhd20};
```



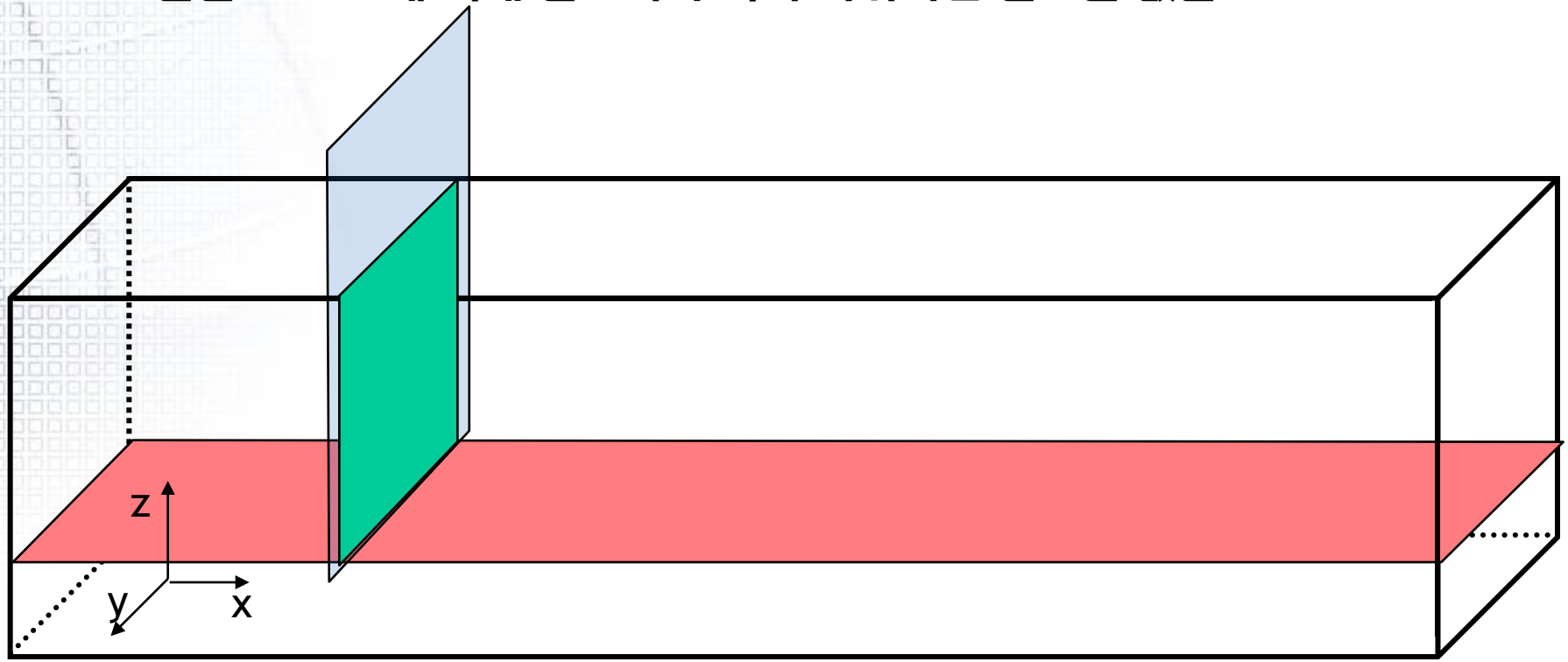
Place 연산: Box 크기에 딱 맞게 X plane을 지정할 필요가 없음

1. 선박 구획 정의
- 구획 모델링의 개념(4)

Place 연산

(Box 안의 구획) Place (X plane 유한 평면) = (Box 안쪽의 평면)

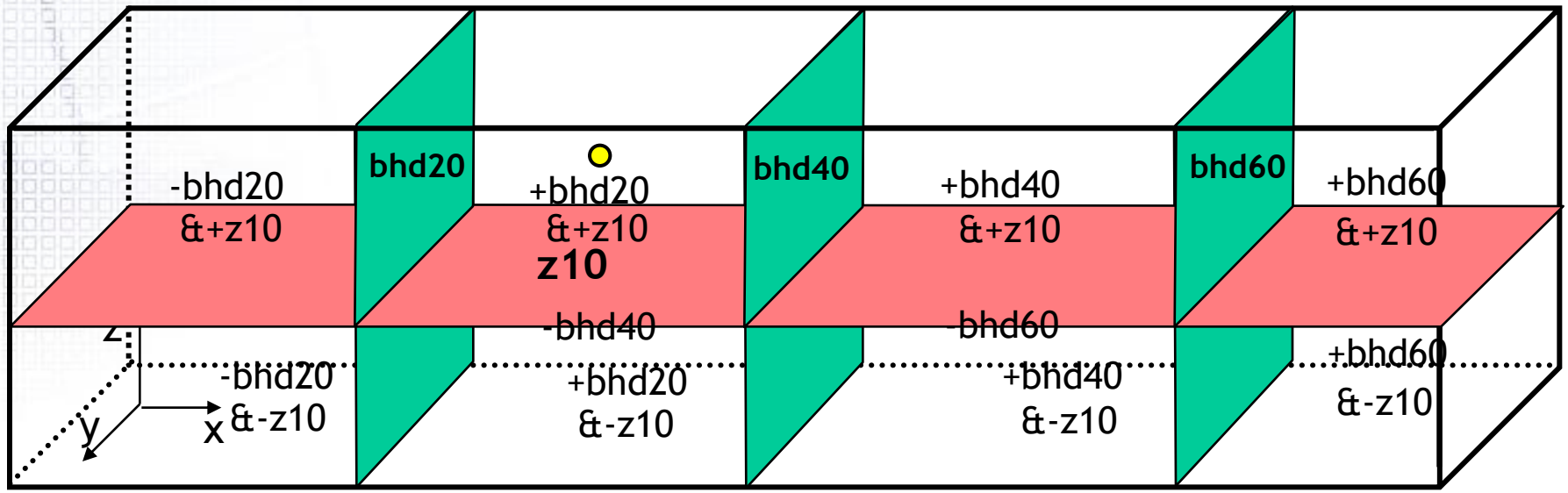
Place 연산: Cutter에 의해 반드시 구획이 나뉘어질 필요는 없음



1. 선박 구획 정의
- 구획 모델링의 개념(5)

구획 지칭 방법

- 3차원 점을 이용한 방법: ~(25, 0, 12)
- Cutter의 (+), (-)로 지정: 해당 구획을 둘러쌓고 있는 cutter만 유용
  “F3” function 키
  화면에서 구획 선택 “cc”명령어

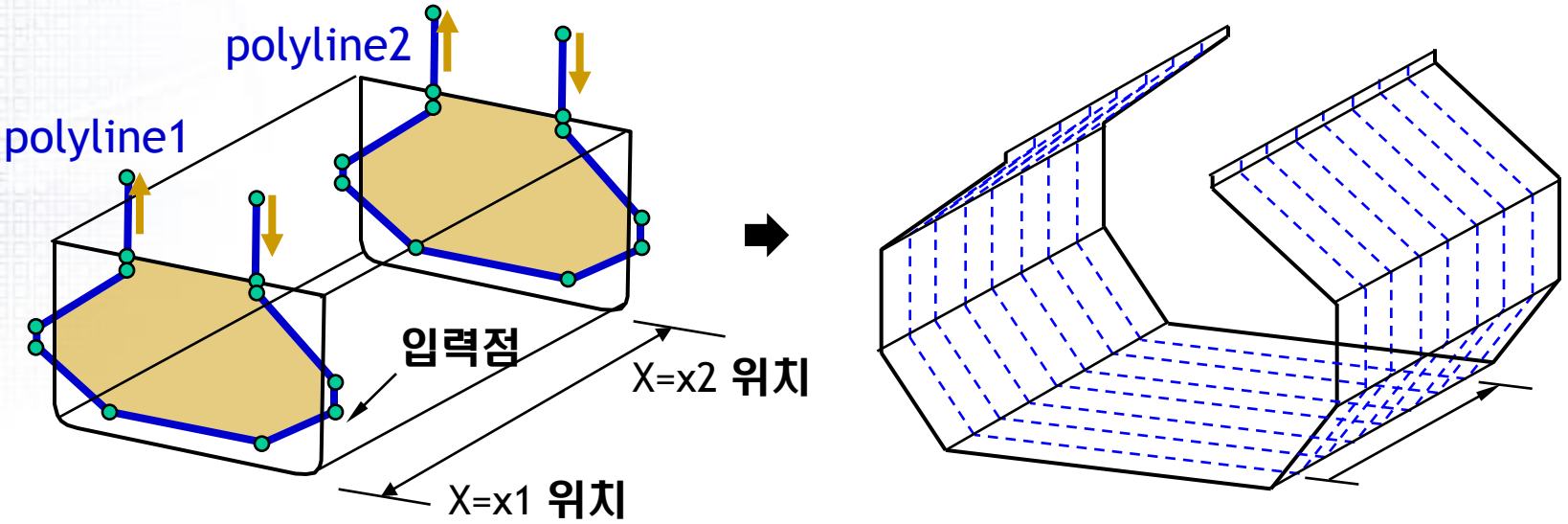


1. 선박 구획 정의
- 구획 모델링의 개념(6)

꺾여진 복잡한 Cutter의 정의

복잡한 분할면을 정의하기 위해서 특정 위치에서의 단면 형상을 Polyline으로 정의하고, 정의된 모든 Polyline을 Skinning하여 생성(skinsurf)할 수도 있다.

skinsurf의 개념: polyline들을 정의 ➡ skinning ➡ surface



✓**주의점: Polyline들의 Point 개수는 동일해야 한다.**

배치 설계 시스템의 구획 정의 파일
- 구획 정의 파일(*.xab)의 전체 구성

*.xab 파일

```
// frame 간격 정의 부분(필수 사항)
xframe ….

// 변수 선언 부분(사용자 편의에 따른 선택 사항)
var …..

// 구획 분할면 모델을 생성할 Box 정의 부분(필수 사항)
box …..

// base 블록 부분: 앞에서 정의된 대상 Box에 대해 분할면 place 연산을 통해 구획 생성 및 이름 정의
base Box 이름 { ….
        ….
        ….
}
```

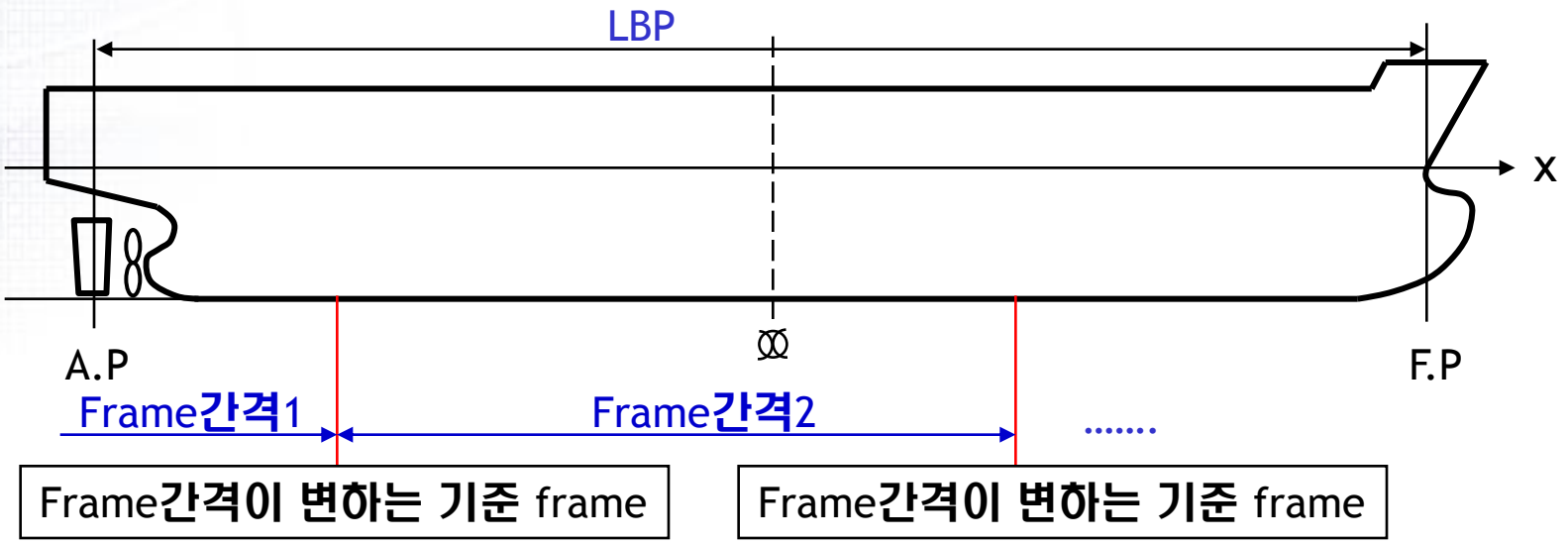
배치 설계 시스템의 구획 정의 파일
- Frame 간격의 정의(필수 사항)

명령어 형식

xframe (frame간격1, 기준frame, frame간격2, ….)

사용 예

xframe (0.85, 60, 5.67, 105, 0.85); //선미에서 60번 frame까지는 0.85m간격,
// 60번 frame에서 105번 frame까지는 5.67m, 105번이상은 0.85m 간격임을 정의



배치 설계 시스템의 구획 정의 파일
- 변수의 선언(선택 사항)

명령어 형식

var **변수 이름** = **변수값**;

사용 예

var Lpp = 320; // 320m**라는 값을 갖는 변수** L **정의**

var er_bhd_frame = 60#x0; //**앞에서 정의한** frame**을 사용하여 변수값을 할당**

// 60**번** frame + 0m**의 값을 변수** er_bhd_frame**에 정의**

- ✓ Frame**에 의한 위치 표시** : (Frame**번호**)#x(**지정**frame**으로부터 떨어진거리** : dx)

  시스템 내부적으로는 앞에서 정의된 frame**간격 정보로부터 지정된** Frame**번호를** meter**단위로 계산하고** dx**만큼의 거리를 더해 결과값을** meter**로 저장한다.**

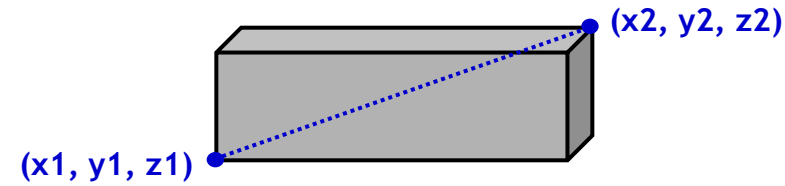
- ✓ **어떤 경우에 변수(**var**명령)를 사용하는가?**

  구획을 정의하기 위해 필요한 좌표값을 직접 사용하기 보다 그 의미가 좀 더 명확할 수 있는 경우에 주로 사용한다. 주로 Main Dimension, **이중저 높이, 주요 격벽 위치, 종격벽의** Knuckle**위치 등의 의미있는 좌표값들에 사용하는 것이 좋으며, 효율적으로 정의되면 나중에 실제 값이 변경되더라도 전체를 일률적으로 변경할 수 있는 장점이 있다.**

배치 설계 시스템의 구획 정의 파일
- Box의 정의(필수 사항)

명령어 형식

box Box **이름** {(x1, y1, z1), (x2, y2, z2)};



사용 예

box ship { (-10, -31, -1), (330, 31, 30) };

// 선박의 전체크기(길이, 폭, 높이)를 포함하는 Box를 정의. 대각선의 두 점을 3차원 좌표값으로 입력. 아래의 예에서와 같이 앞에서 정의된 변수와 변수들의 사칙연산을 이용하여 정의할 수도 있음.

box ship {(-aft_dist -tot, -B/2 -tot, -1), (Lpp+fore_dist+2, B/2+tot, D -0.01)};

✓ **변수들의 연산으로 좌표를 표현할 경우 주의점: 뺄셈("-")연산자**

변수이름 작성 규칙이 변수의 이름 중간에 "-"문자를 사용할 수 있도록 되어있으므로 연산으로 좌표를 표현하는 경우, 변수가 먼저 오는 뺄셈("-")연산자의 경우에는 반드시 공백을 한 칸 넣어주어야 함.(앞의 예에서, -aft_dist -tot, -B/2 -tot, D -0.01))

배치 설계 시스템의 구획 정의 파일 - “base” 블록의 명령어 리스트

“base” 블록 내의 명령어 리스트

명령어	기능 설명
base	대상 모델을 지정, 대부분 구획 분할면 모델을 위해 Box로 정의한 모델을 지칭
xplane	x축에 수직인 평면 생성, 구획의 횡격벽에 주로 사용
yplane	y축에 수직인 평면 생성, 구획의 종격벽에 주로 사용
zplane	z축에 수직인 평면 생성, Deck이나 Floor에 주로 사용
polyline	Polyline생성, 복잡한 형상의 화물창을 정의할 때 skinsurf와 함께 사용
skinsurf	여러 Polyline을 지나도록 Skinning하여 면 생성, 복잡한 형상의 화물창 정의 시 사용
reflectsurf	대상 면의 대칭인 면 생성, Starboard만 정의하고 Port쪽 구획을 정의할 때 사용
union	Boolean Operation. 모델 합집합 연산, 선체 부가물 정의 등에 사용
subtract	Boolean Operation. 모델 차집합 연산, Bow Thruster정의 시 사용
intersect	Boolean Operation. 모델 교집합 연산
place	Boolean Operation을 확장한 Region분할 연산, 구획을 만들기 위한 필수 명령
load	파일로부터 모델을 메모리상으로 Load, 선형다면체모델이 주 대상임
defname	구획의 이름 정의

- ✓ base 블록의 의미: base Box 이름 {.....}

선형과 독립된 구획 모델을 만들기 위해 EzCOMPART에서는 선형을 포함하는 크기의 Box를 먼저 앞 페이지의 box명령에 의해 생성하고, 그 Box를 대상으로 내부 구획 분할면을 정의하고 place연산에 의해 구획을 나누는 과정을 거친다. Base명령은 대상 Box가 어떤 것인지를 지칭하며, 위의 표에 있는 base블록내의 명령어 리스트는 모두 base에 의해 지칭된 대상 Box에 대한 명령으로써 동작한다.

배치 설계 시스템의 구획 정의 파일 - “base” 블록의 전체 구성

“base” 블록

base Box **이름** {

// 구획 분할면 정의 부분

xplane(yplane, zplane) ….

skinsurf ….

reflectsurf ….

// 연산에 의한 구획 정의 부분

place ….

union ….

subtract ….

intersect ….

// 다른 모델 열기

load ….

// 구획 이름 정의 부분

defname ….

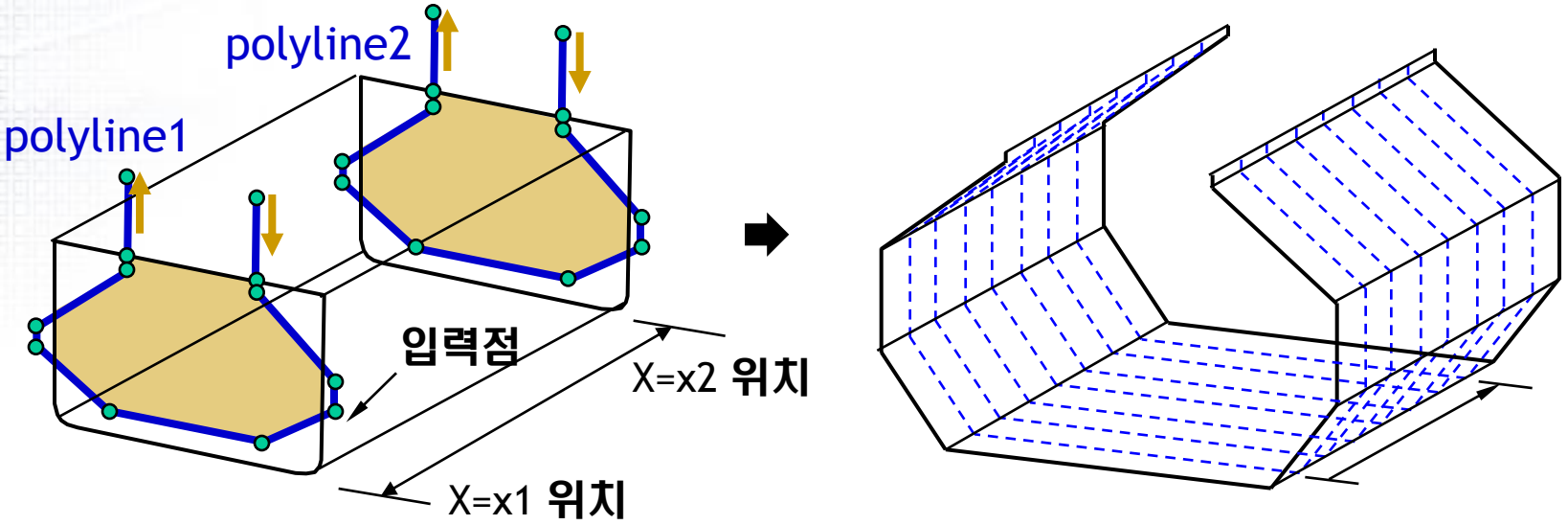
}

배치 설계 시스템의 구획 정의 파일 - “base” 블록: 구획 분할면의 정의(1)

구획 분할면 정의 부분

여러 가지 면 정의 방법을 제공하며, 각각의 정의 방법에 따라 하나의 명령줄에서 면을 정의(xplane, yplane, zplane, reflectsurf)할 수도 있고, 복잡한 분할면을 정의하기 위해서 특정 위치에서의 단면 형상을 Polyline으로 정의하고, 정의된 모든 Polyline을 Skinning하여 생성(skinsurf)할 수도 있다.

skinsurf의 개념 : polyline들을 정의 ➡ skinning ➡ surface



✓주의점: Polyline들의 Point 개수는 동일해야 한다.

배치 설계 시스템의 구획 정의 파일
- “base” 블록: 구획 분할면의 정의(2)

명령어 형식

xplane **이름** x**위치값**;
yplane **이름** y**위치값**;
zplane **이름** z**위치값**;
reflectsurf **이름** **좌표**(x,y,z)=**값** **대상면**;

사용 예

```
xplane er_bhd  er_bhd_frame;  // er_bhd_frame 변수 위치에 er_bhd 이름의 x 평면 생성
yplane  center_bhd  0;        // y=0 위치에 center_bhd 이름의 y 평면 생성
zplane  db  2.0;              // z=2.0 위치에 db 이름의 z 평면 생성
reflectsurf  rss1s  y=0  ss1s;  // y=0 평면을 기준으로 이미 정의된 ss1s면을 대칭시켜
                                // rss1s면을 생성
reflectsurf  rss1s  x=10  ss1s;  // x=10 평면을 기준으로 이미 정의된 ss1s면을 대칭시켜
                                 // rss1s면을 생성
```

배치 설계 시스템의 구획 정의 파일
- "base" 블록: 구획 분할면의 정의(3)

명령어 형식

pointlist2d **이름** [**반복회수**~(dx,dy)] {(x1, y1), PLstID, [**반복회수**~(dx2, dy2)]{(x2, y2), ~(dx2, dy2), … }, … }

polyline **이름** **좌표**(x | y | z)=**값** PLstID | {(y1,z1),(y2,z2), …}; //**좌표선택을** x**로 한 경우**

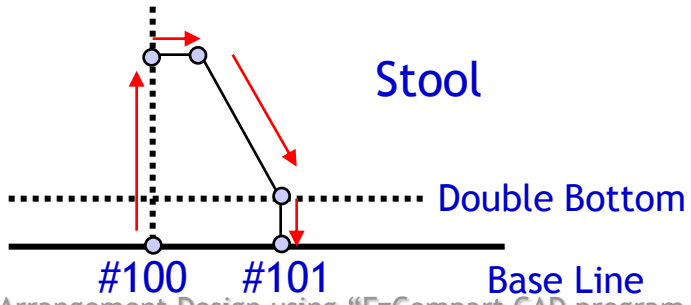
skinsurf **이름** {polyline1, polyline2,….};

사용 예

polyline pl1c x=er_bhd_frame { (-6.37, up_max), (-6.37, db_height), (6.37, db_height), (6.37, up_max)}; //x=er_bhd_frame **단면 위치에** (-6.37, up_max),
//(-6.37, db_height), (6.37, db_height), (6.37, up_max)**의** 4**개의** (y,z) **좌표점을**
//**지나는** polyline**인** pl1c**를 정의**.

skinsurf ss1c {pl1c, pl2c, pl3c, pl4c}; //**기 정의된** polyline pl1c, pl2c, pl3c, pl4c**를**
//**지나는** skinning**면 정의**.

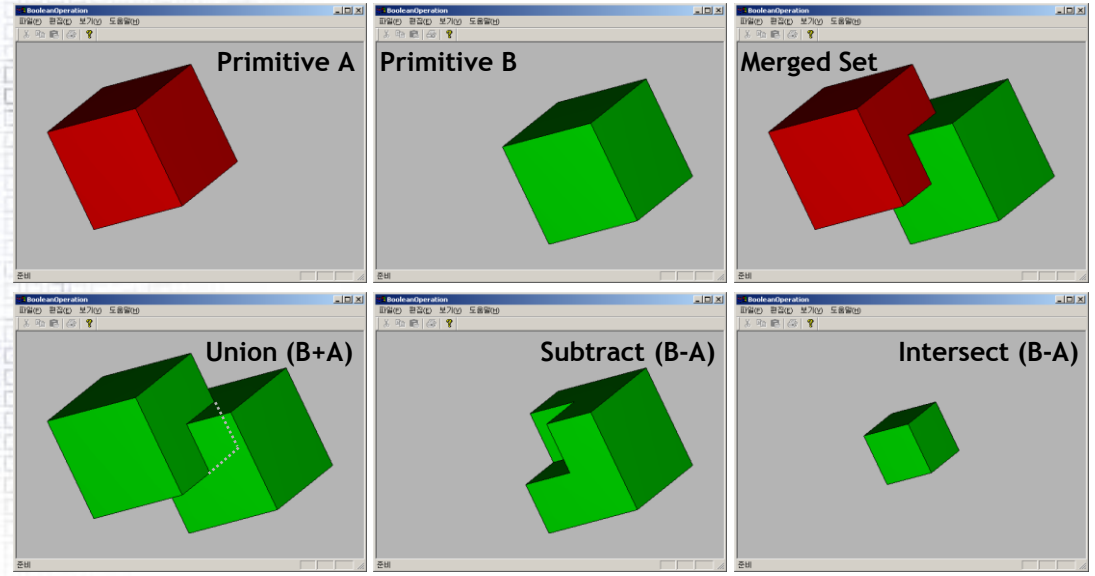
polyline aa y=0 {(100#x0, 0), (100#x0, 6), (100#x0.8, 6), (101#x0, 2), (101#x0, 0)};
// y=0**평면에서** 5**개의** (x,z) **좌표점을**
// **지나는** Stool **형상을 정의**



배치 설계 시스템의 구획 정의 파일 - “base” 블록: 연산에 의한 구획 정의(1)

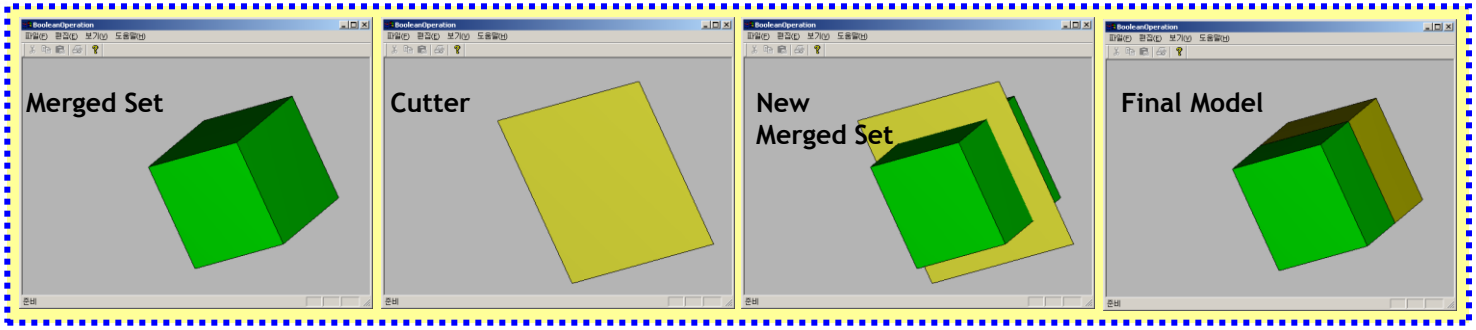
연산에 의한 구획 정의 부분

앞서 정의된 구획 분할면을 이용하여 기존 구획을 분할할 수 있는 확장된 구획 정의 연산인 place를 제공하여 대부분 구획을 생성하고, 솔리드 모델간의 기본적인 Boolean연산(합집합-union, 교집합-intersect, 차집합-subtract)도 제공한다.



Boolean Operations

Place Operation



배치 설계 시스템의 구획 정의 파일 - “base” 블록: 연산에 의한 구획 정의(3)

명령어 형식

place (분할면1, 분할면2, ….) @{±분할면1, ±분할면2, …} // 부록: 구획 지정 참조

사용 예

place (er_bhd) @{*}; //base로 지정된 대상 구획에 대해 분할면 er_bhd로 분할

// 선박을 Engine Room구역과 전체 화물창 구역으로 나누는 작업

place (no1_bhd, no2_bhd, no3_bhd, no4_bhd) @{+er_bhd};

// 분할면 er_bhd의 +방향에 있는 구획에 대해 no1_bhd, no2_bhd, no3_bhd,

// no4_bhd의 분할면으로 place연산한다. 화물창 구획을 횡 격벽으로 하나씩

// 분할하여 no.1 cargo ~ no.5 cargo 구획으로 5개의 구획을 나누는 작업

place (slop_bhd) @+er_bhd{-ss1s, -rss1s};

// 분할면 er_bhd의 +방향에 있는 구획과 분할면 ss1s의 -방향에 있는 구획의

// 교집합 구획에 대해 slop_bhd의 분할면을 place연산하고, rss1s의 -방향에 있는

// 구획의 교집합 구획에 대해서도 place 연산한다.

// place (slop_bhd) @{+er_bhd&-ss1s, +er_bhd&-rss1s}; 와 동일한 개념

배치 설계 시스템의 구획 정의 파일 - “base” 블록: 연산에 의한 구획 정의(4)

Place 연산자에 의한 모델링 예 1

box Box **이름** {(x1, y1, z1), (x2, y2, z2)};

ex) box b1 { (-aft_dist, -B/2, -1), (Lpp+fore_dist , B/2, D) };

⇩

xplane **이름** x **위치값**;

ex) xplane er_bhd 51;

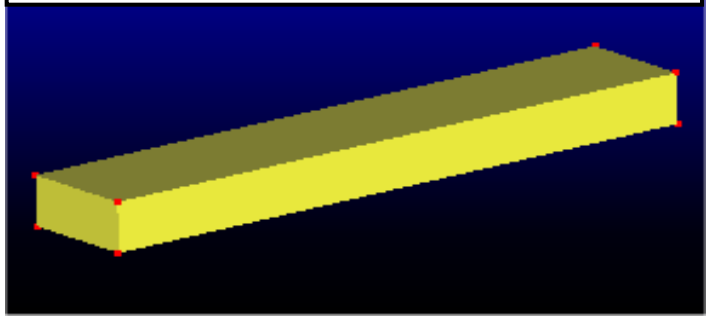
⇩

place (**분할면**1, ….) @{±**분할면**1, ±**분할면** 2,…};

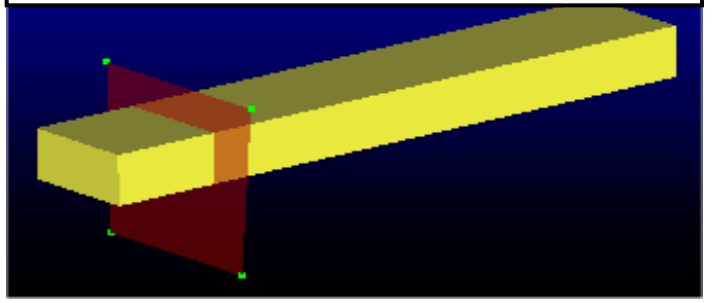
ex) place (er_bhd) @{*};

✓ 주의점: 개념을 형상화하기 위해 “base” 블록 부분 생략

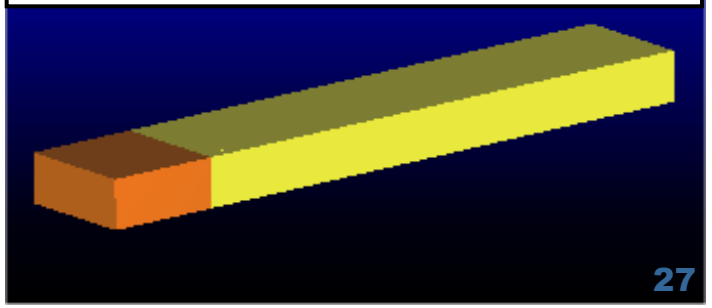
주요 치수로 box b1을 생성



x=51m에 무한 평면 er_bhd 생성

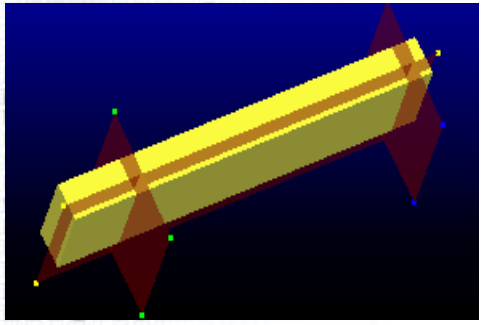


b1이 er_bhd면에 의해 두 구획으로 분할



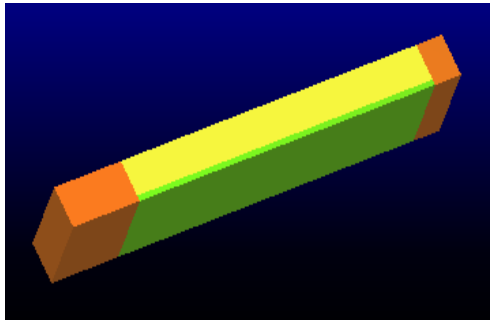
배치 설계 시스템의 구획 정의 파일 - “base” 블록: 연산에 의한 구획 정의(5)

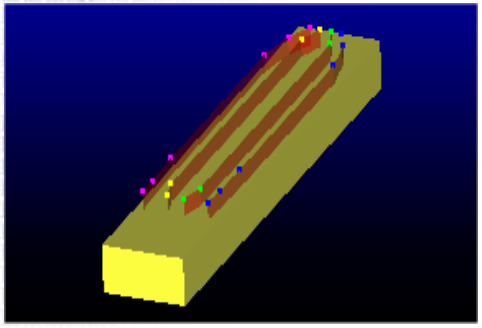
Place 연산자에 의한 모델링 예 2



```
place (er_bhd, coll_bhd) (*);
place (db) @{+er_bhd};
```

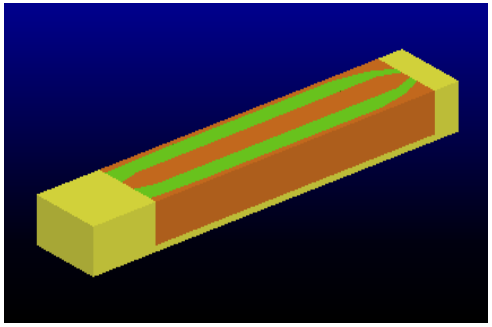
box b1을 er_bhd, coll_bhd로 나눈 후
er_bhd의 +방향 구획 만을 db로 나눈다
* 구획수 4개 (선미부, 선수부, 화물창부, 이중저)

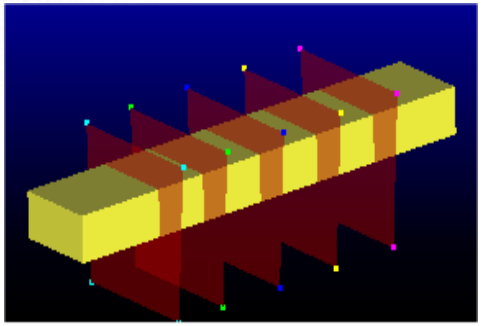




```
place (ss, sc, ps, pc) @{+er_bhd&+db};
```

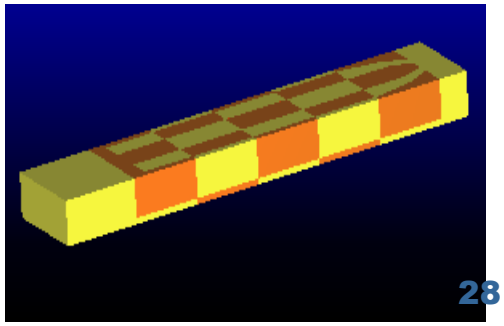
er_bhd의 +방향이자, db의 +방향인 구획 (화물창부)을 ss, sc, ps, pc 면으로 나눈다
* 화물창부가 5개의 구획으로 나뉘어 짐





```
place (bhd1, bhd2, bhd3, bhd4) @{+er_bhd};
place (slop) @{+er_bhd&+sc, +er_bhd&+pc};
```

er_bhd의 +방향인 구획들(화물창부 5개 , 이중저)을 한꺼번에 bhd1, bhd2, bhd3, bhd4 면으로 나눈 후, er_bhd의 +방향, sc의 +방향인 구획과, er_bhd의 +방향, pc의 +방향인 구획(종 화물창부 2개)을 slop면으로 나눈다



배치 설계 시스템의 구획 정의 파일
- “base” 블록: 모델의 Loading(1)

모델의 Loading 부분

복잡한 형상의 모델인 경우, 미리 다른 파일에 저장한 후 구획정의 파일(*.xas)에서 메모리 상에 load하여 여러 연산을 적용할 수 있다. 선형 모델을 구획 정의 파일에서 Loading하여 Place연산하고자 하는 경우에 주로 사용한다.

명령어 형식

load **모델 이름** [/d] **“파일 경로 및 이름”** ;

사용 예 :

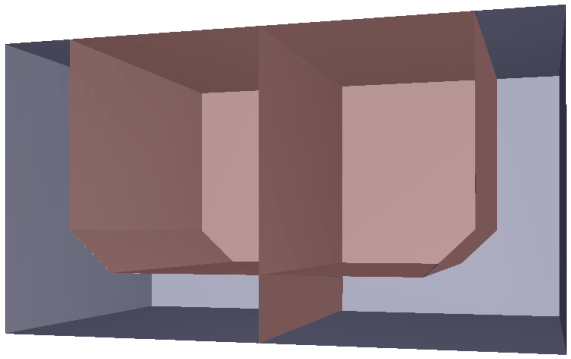
```
load hull /d "VLCC320_s_union_mesh_2000.xac";
    //현재 디렉토리에 있는 “VLCC320_s_union_mesh_2000.xac”라는 파일의 내용을
    // hull이라는 이름의 모델로 메모리 상에 올린다. 단, 여기서 /d의 의미는 place연산을
    // 수행할 때, 선체외곽부분은 모두 삭제(delete)하겠다는 의미
place (hull) @{*};    // Box로 구성한 구획 분할면 모델을 선형 모델로 Place 연산하고,
    // 선체 외곽 부분은 모두 삭제하여 완전한 구획 모델이 완성된다.
```


✓ *.xac 파일 형식: 배치 설계 시스템으로부터 생성된 모델을 저장하고 있는 Binary파일 형식

배치 설계 시스템의 구획 정의 파일
- “base” 블록: 모델의 Loading(2)

모델의 Loading 부분

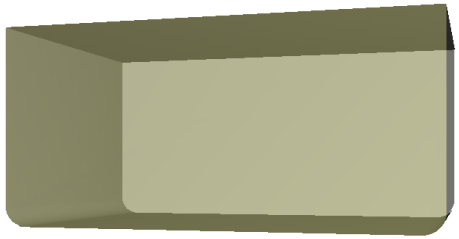
Subdivision BHD/Deck **모델**



+

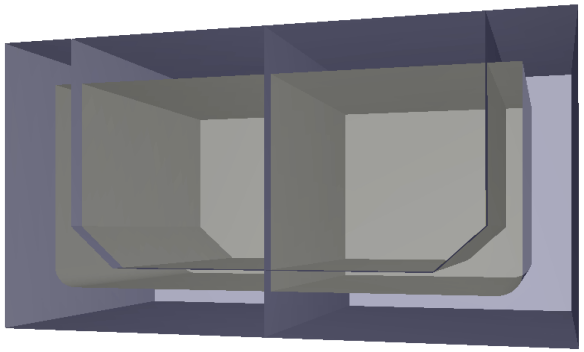
Hull **모델** Loading

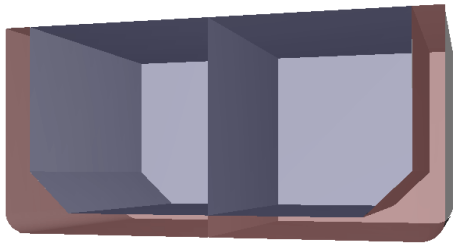
load hull /d "VLCC320_s_union_mesh_2000.xac";



Place

병합체(Merged Set)





배치 설계 시스템의 구획 정의 파일 - “base” 블록: 구획의 이름 정의

구획의 이름 정의

배치 설계 시스템의 구획모델링 개념은 구획 분할면을 정의하여 전체 선체모델의 내부를 분할해가며 구획모델을 완성해 가는 과정으로 이루어지므로, 구획 분할면 정의가 완료되면 구획 모델이 완성된다. 다만 각 구획의 이름은 최종 완성된 구획에 대해 부여하여야 하므로 최종 구획 완성 단계에서 구획 이름을 정의하는 절차를 밟는다.

명령어 형식

defname “구획 이름” @{±분할면1, ±분할면2, ...};

사용 예 :

defname "F.P. TK" @{-fp_top};

// 분할면 fp_top의 -방향에 있는 구획의 이름을 “F.P.TK”로 정의한다.

defname "W.B. TK(P) No. * " @{+no1_bhd, +no2_bhd, +no3_bhd, +no4_bhd};

// 별표(“*”)의 의미는 분할면 no1_bhd, no2_bhd, no3_bhd, no4_bhd 의 +방향에

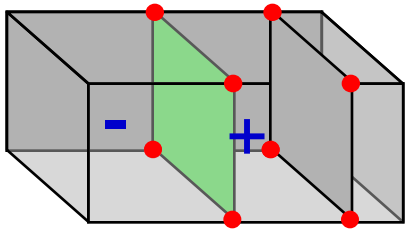
// 있는 각각의 구획들을 순서대로 1부터 Numbering하여 이름을 정의한다.

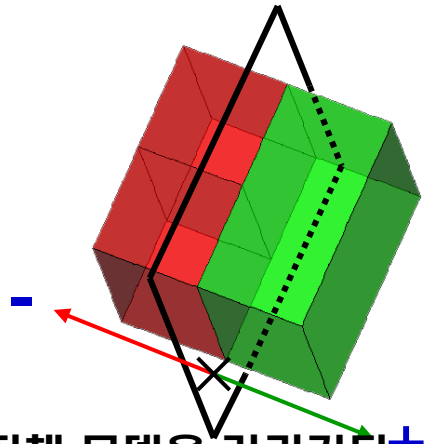
// W.B.TK(P) No.1, W.B.TK(P) No.2, W.B.TK(P) No.3, W.B.TK(P) No.4

배치 설계 시스템의 구획 정의 파일 - 구획의 지정 방법

구획 지정 방법

배치 설계 시스템의 구획 정의 파일(*.xas)에서 생성된 구획을 지정하는 방법은 구획 분할면을 기준으로 +방향, -방향으로 지정한다. 이때 +방향(혹은 -방향)에 있는 모든 구획을 지정하는 것이 아니라 바로 분할면 앞에 있는 구획만 지정한다. 구획 지정이라는 의미를 부여하기 위해 @문자를 함께 사용한다. 또한 분할면의 +혹은 -방향으로 2개이상의 구획이 지정될 수 있으므로 &(교집합) 연산자를 함께 사용하여 특정 구획을 지정할 수 있다.





명령어 형식

@{±분할면1, ±분할면2, ...};
@{±분할면1&±분할면2, ...};

사용 예

place (er_bhd) @{*}; //특별히 "*" 문자는 "base"로 지정된 전체 모델을 가리킨다.

place (no1_bhd, no2_bhd, no3_bhd, no4_bhd) @{+er_bhd};

place (slop_bhd) @{+er_bhd&-ss1s, +er_bhd&-rss1s};

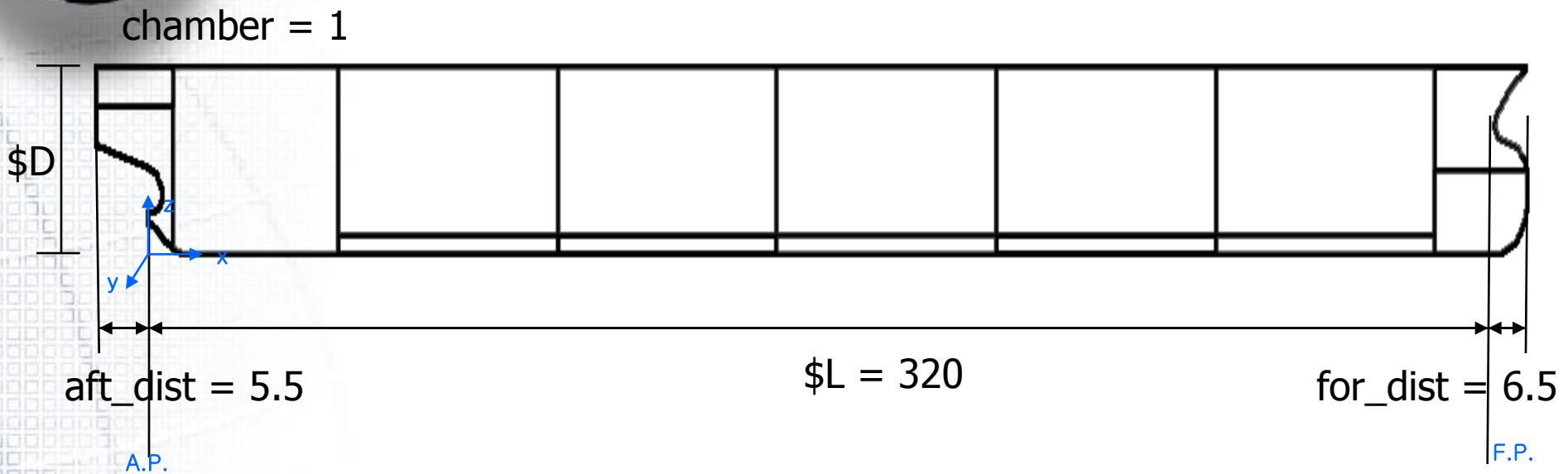
// 위와 같이 동일한 +er_bhd 구획에 대해 교집합 연산을 하는 경우,

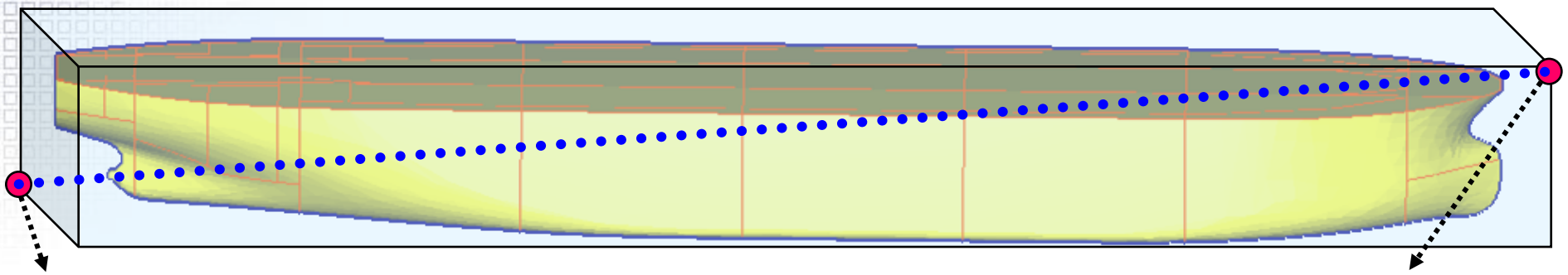
// place (slop_bhd) @+er_bhd{-ss1s, -rss1s}; 와 같이 공통 인자를 뽑아 표현할 수

// 있다.

VLCC의 구획 모델링을 위한 Box의 정의 예

3,700 TEU Container Carrier로 대체





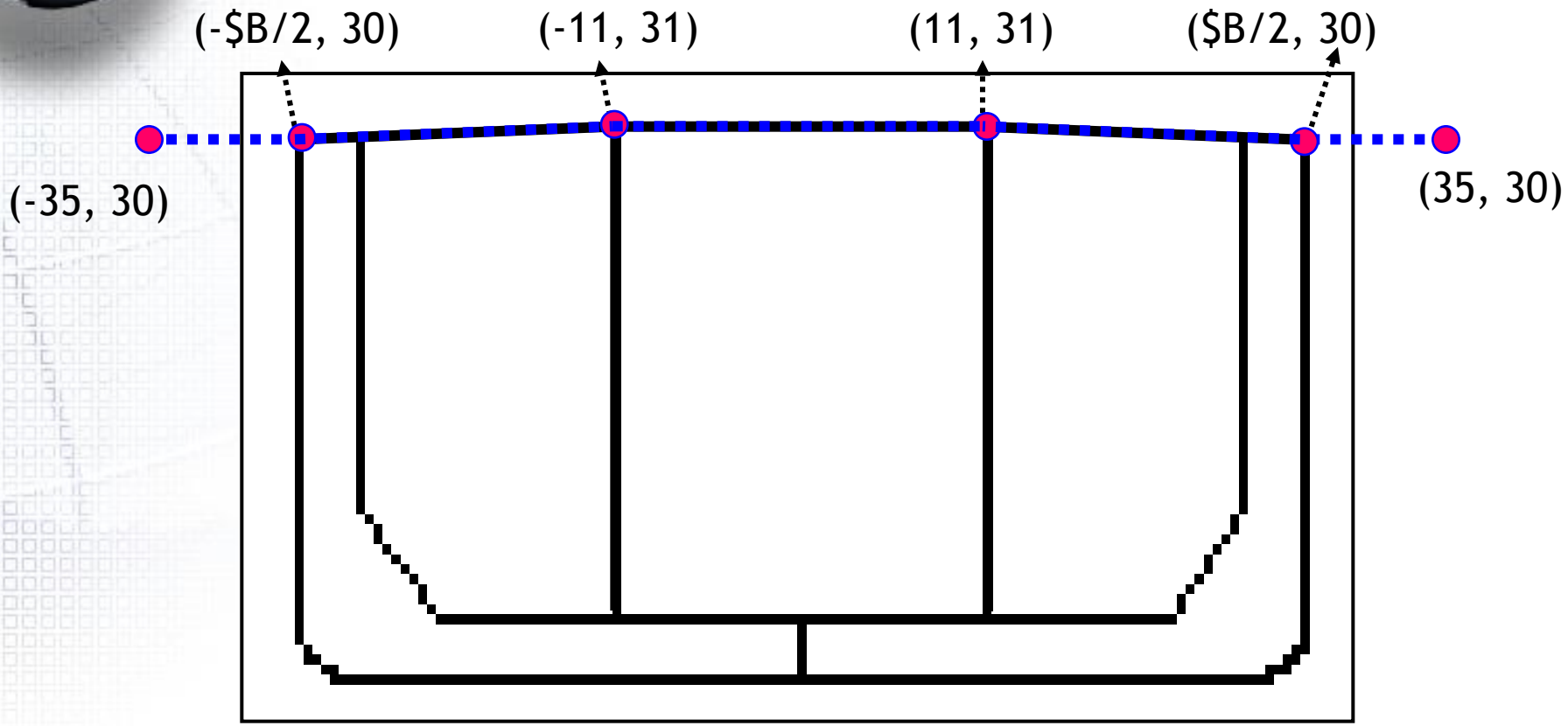
(- aft_dist - tol, - $B/2 - tol, - tol)

($L + for_dist + tol, $B/2 + tol, up_max)

```
box ship /e { (- aft_dist - tol,  - $B/2 - tol, - tol), ($L + for_dist + tol, $B/2 + tol, up_max) }
```

VLCC의 구획 모델링을 위한 Deck의 정의 예(1)

3,700 TEU Container Carrier로 대체



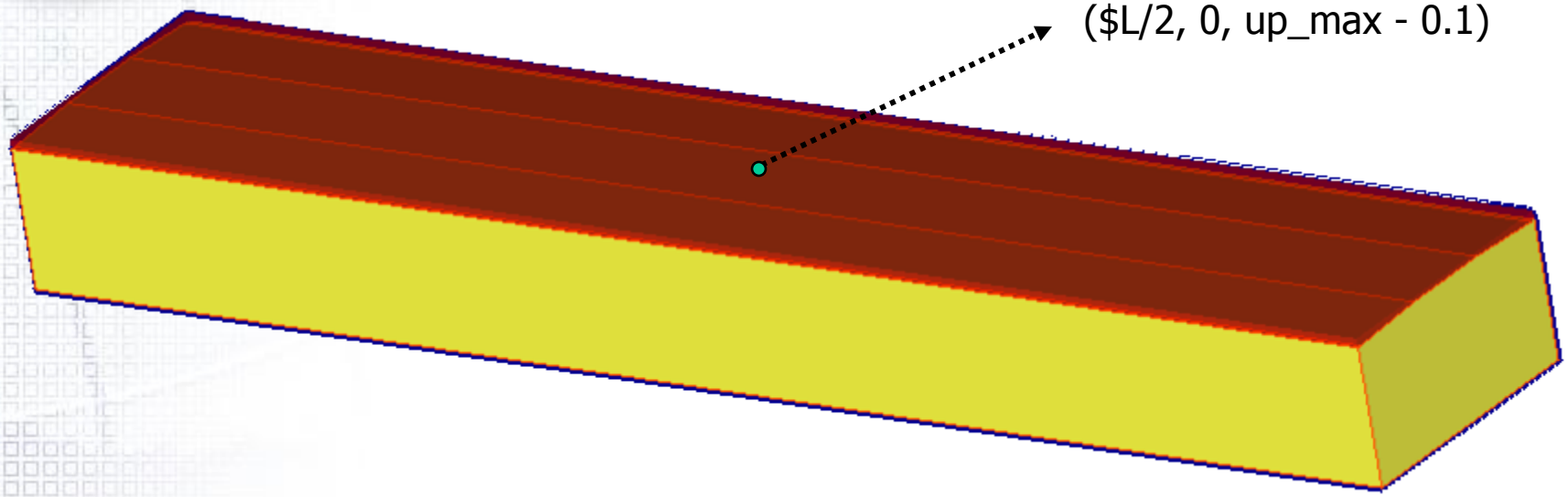

```
polyline polyln1 x=-10 { (-35, 30), (-$B/2, 30), (-11, 31), (11, 31), ($B/2, 30),(35, 30) };
polyline polyln2 x=340 { (-35, 30), (-$B/2, 30), (-11, 31), (11, 31), ($B/2, 30),(35, 30) };
skinsurf deck { polyln1, polyln2 };
place (deck) @{*}
```

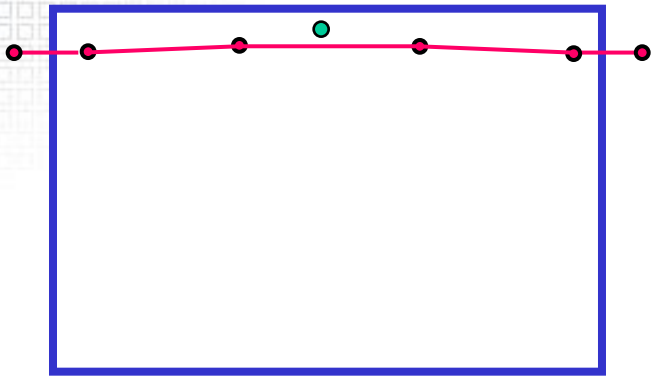
VLCC의 구획 모델링을 위한 Deck의 정의 예(2) - Deck 윗 부분의 삭제

3,700 TEU Container Carrier로 대체

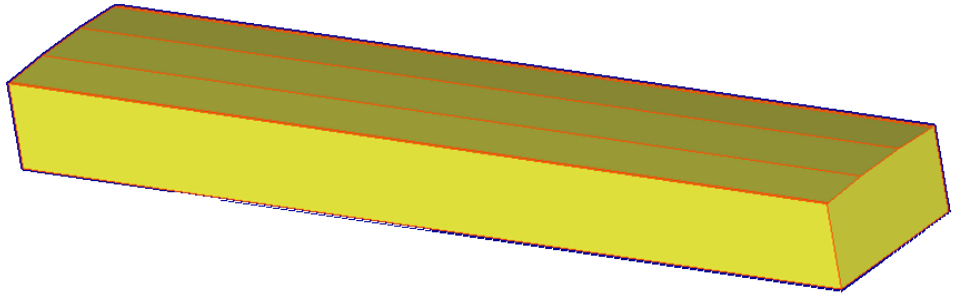
삭제하고자 하는 구획 내부의 점

($L/2, 0, up_max - 0.1)



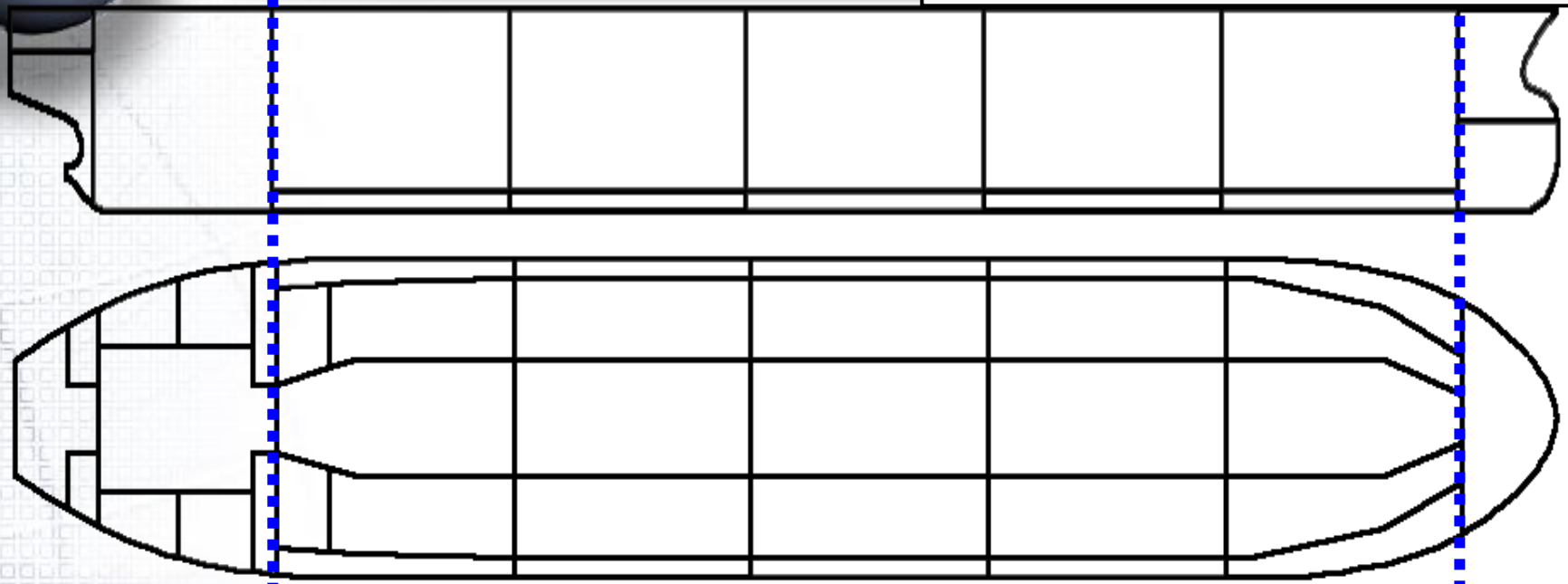


delete @{ ~($L/2, 0, up_max - 0.1) }



VLCC의 구획 모델링을 위한 선미, 중앙 평행부, 선수부 격벽의 정의 예

3,700 TEU Container Carrier로 대체

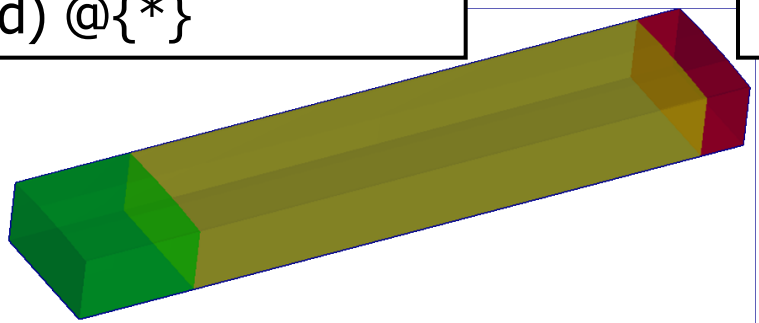


er_bhd_frame = 60#x0

coll_bhd_frame = 105#x0;

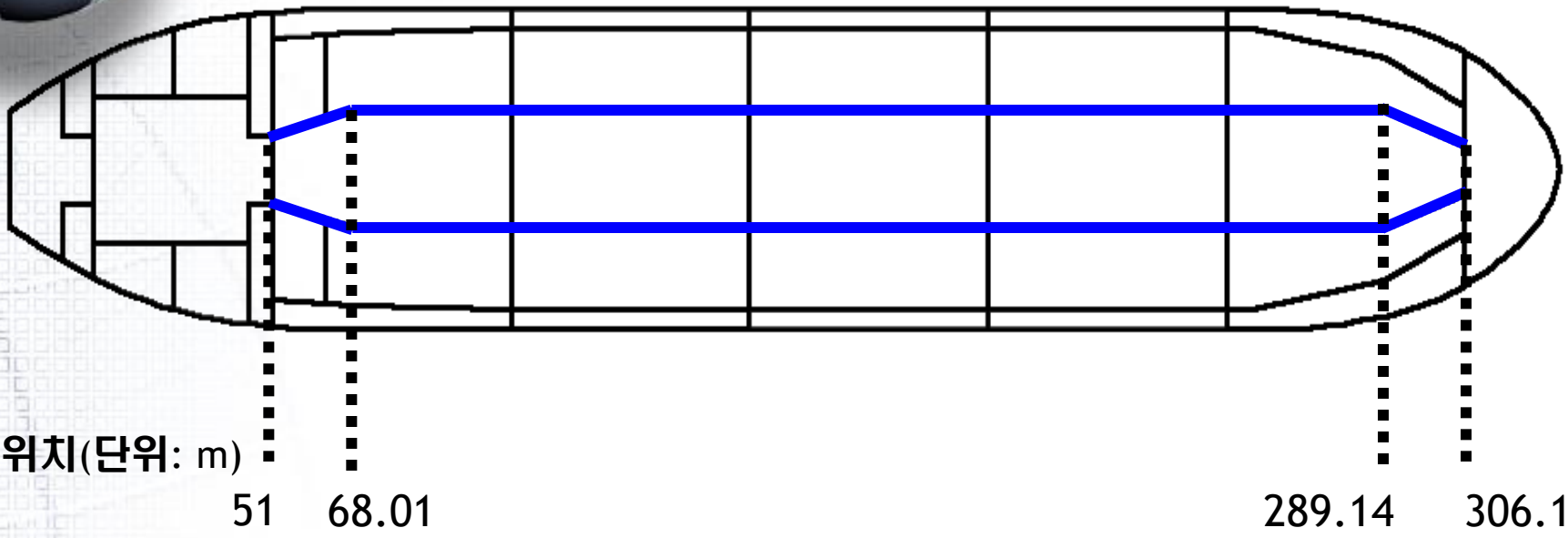
```
xplane er_bhd er_bhd_frame;
place (er_bhd) @{*}
```

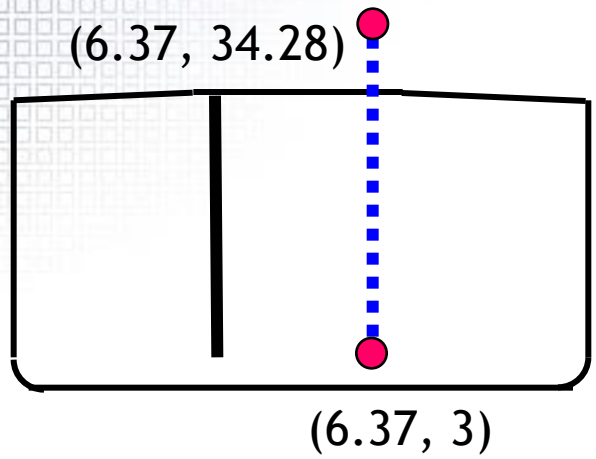
```
xplane coll_bhd coll_bhd_frame;
place (coll_bhd) @{*}
```



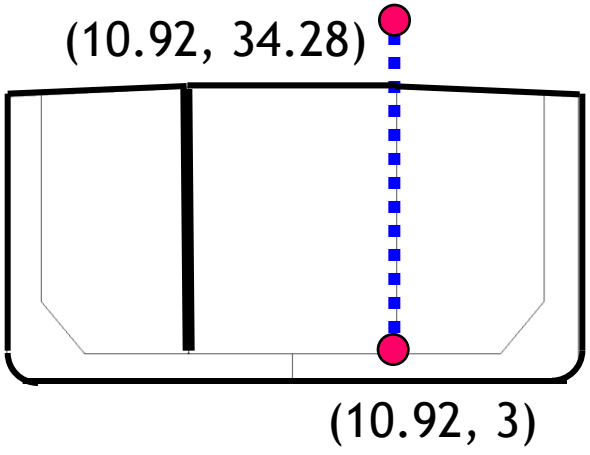
VLCC 구획 모델링을 위한 Longitudinal Bulkhead의 정의 예

3,700 TEU Container Carrier로 대체

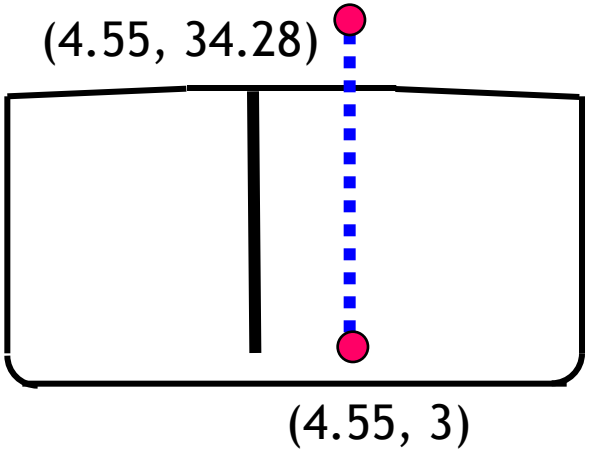




x =51



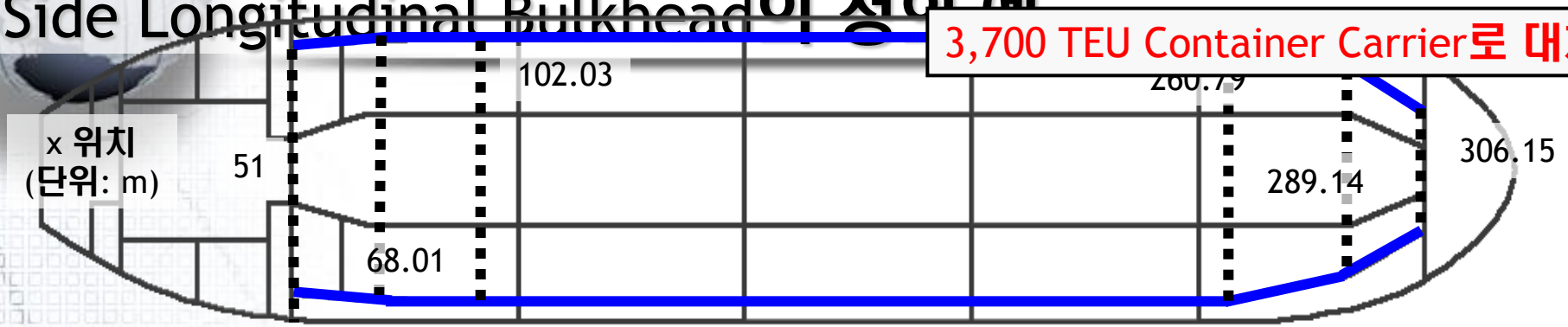
x =68.01, 289.14

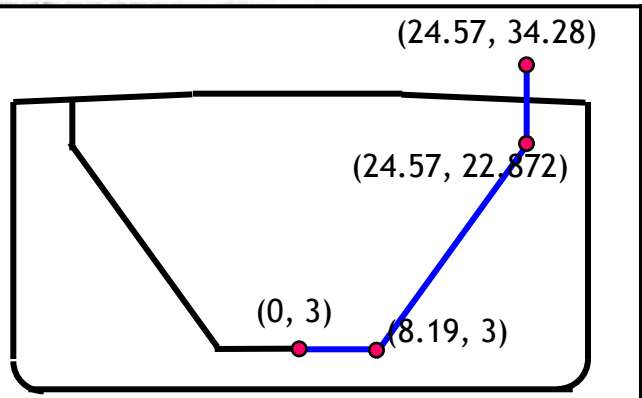


x =306.15

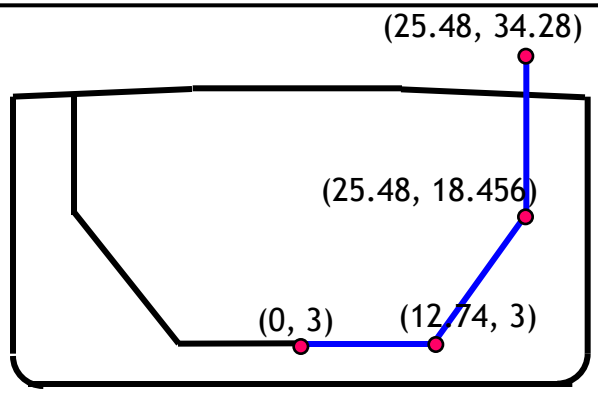
VLCC의 구획 모델링을 위한 Side Longitudinal Bulkhead의 정의 예

3,700 TEU Container Carrier로 대체

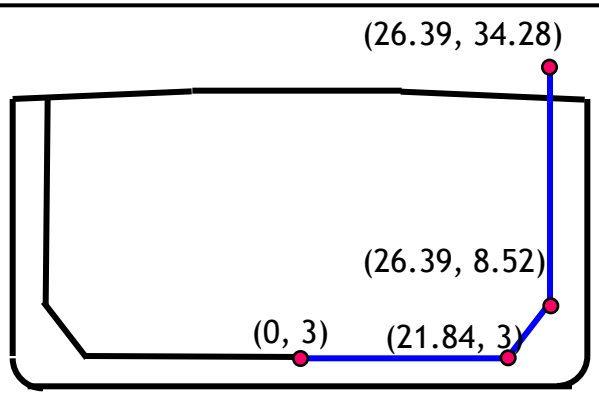




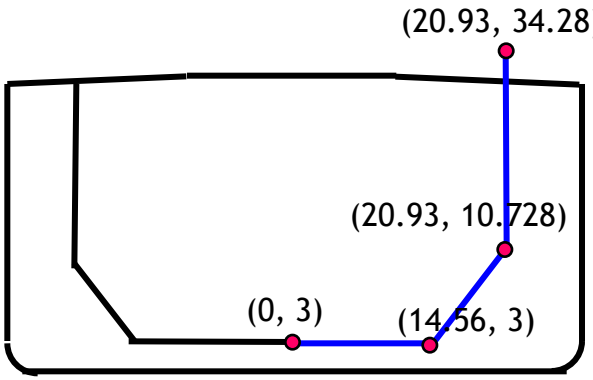
X =51



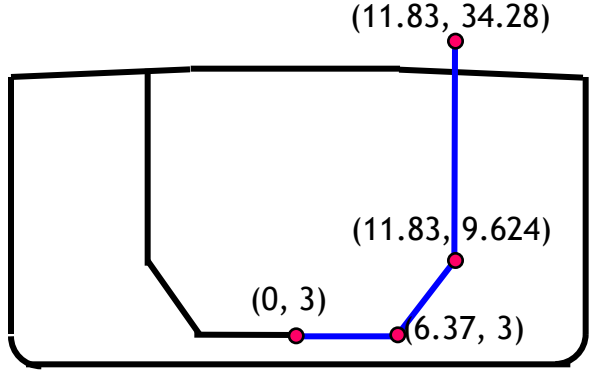
X =68.01



X =102.03, 260.79



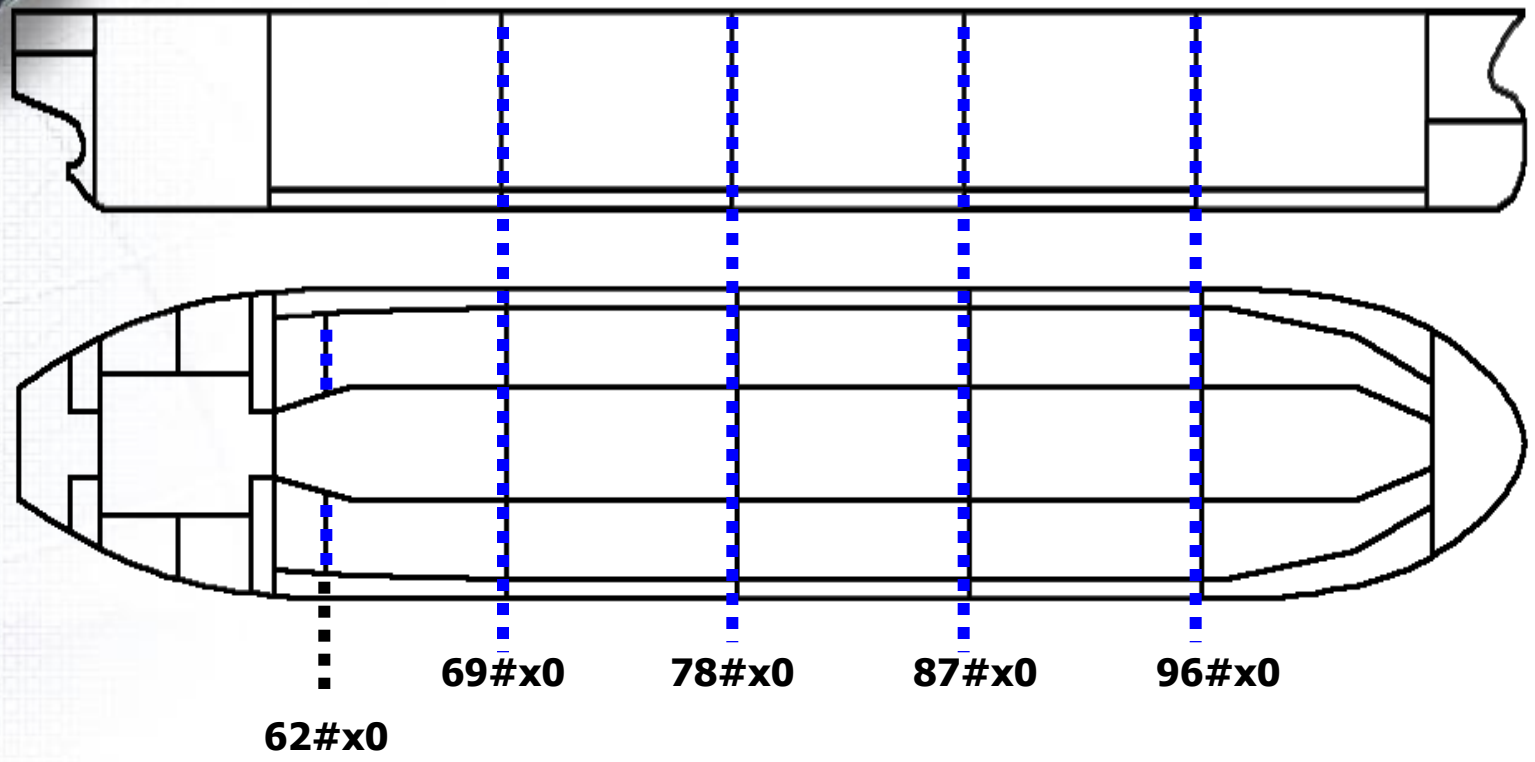
X =289.14

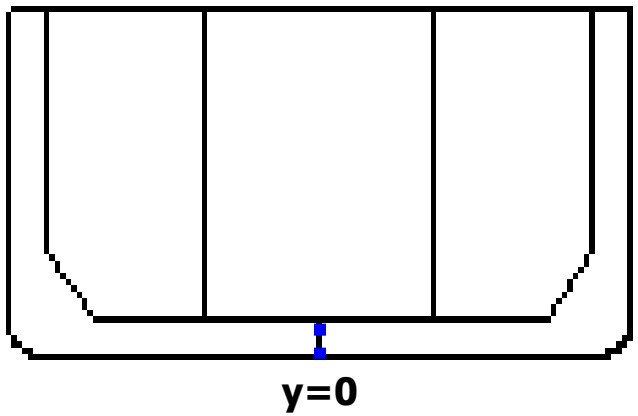


X =306.15

VLCC의 구획 모델링을 위한 Transverse Bulkhead와 Center Girder의 생성 예

3,700 TEU Container Carrier로 대체

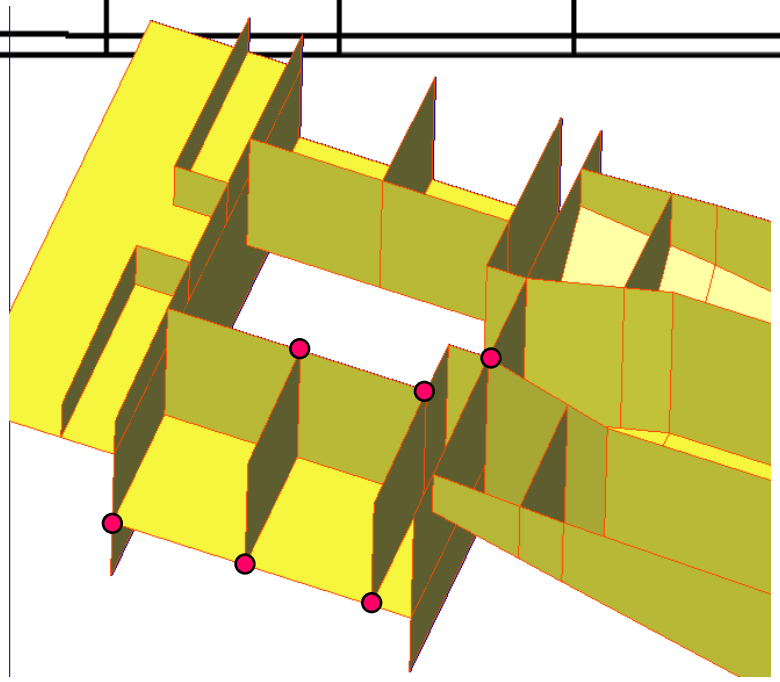


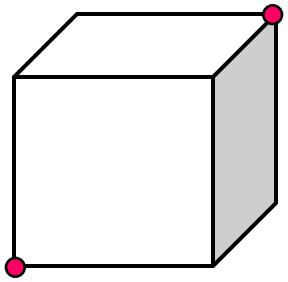


중앙 평행부의 Cargo Tank 아래 부분

VLCC의 구획 모델링을 위한 선수/선미부의 정의 예

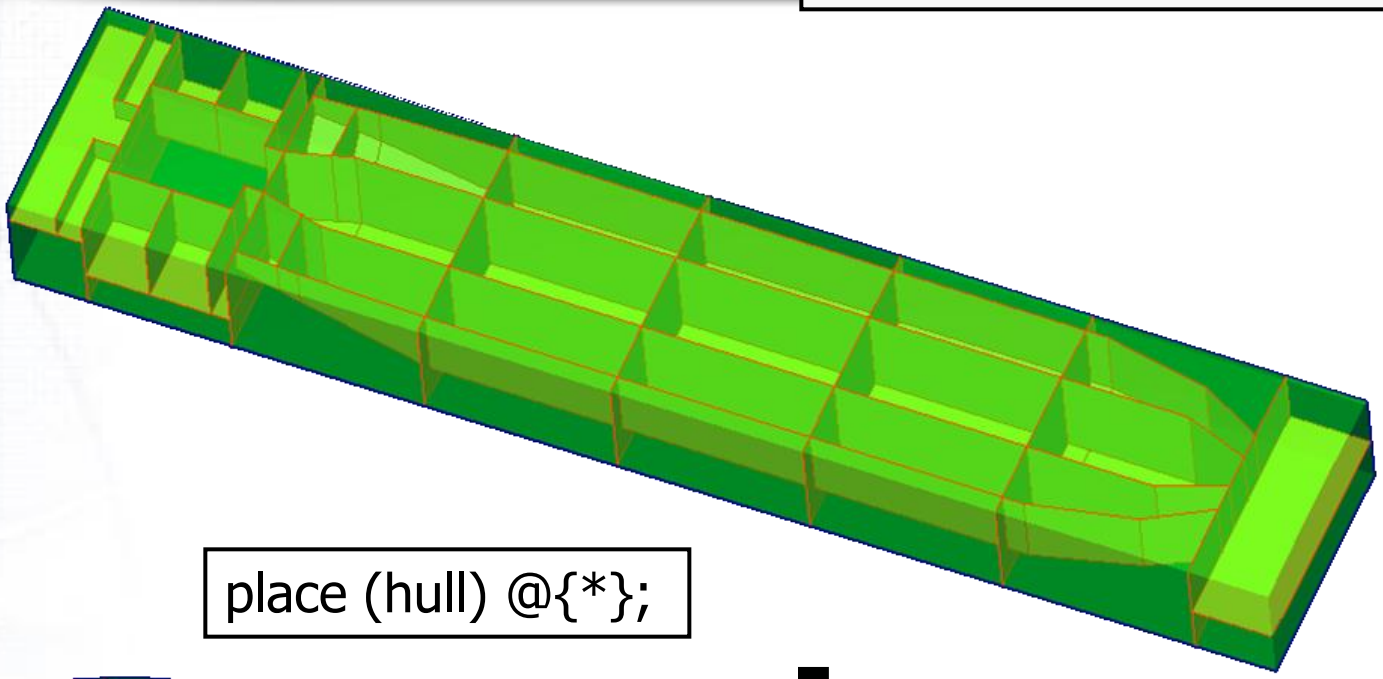
3,700 TEU Container Carrier로 대체



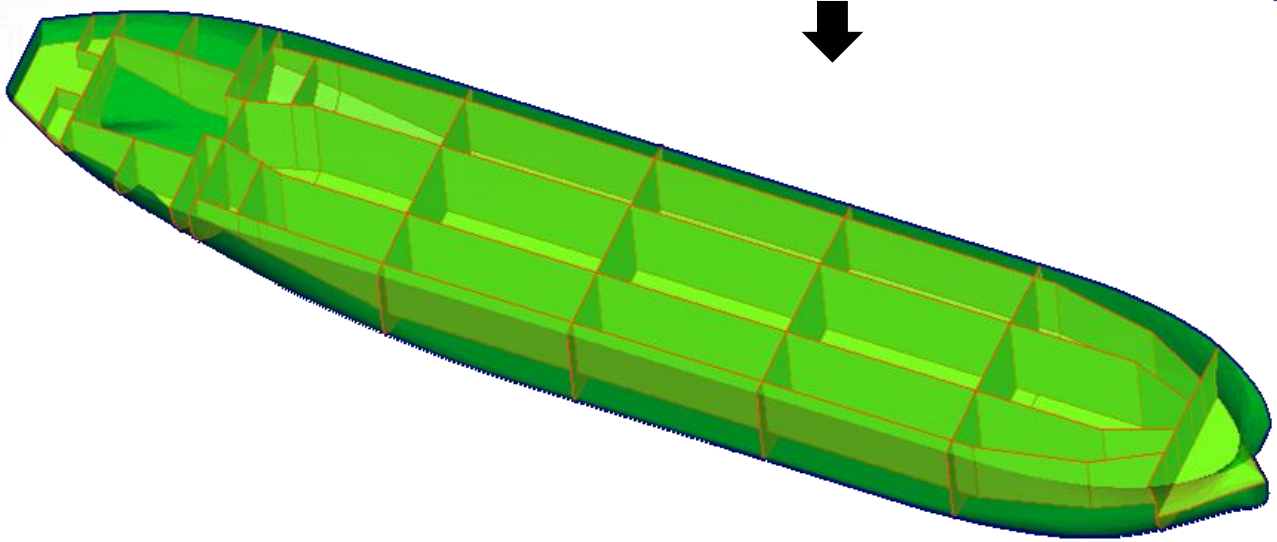


VLCC의 구획 모델링을 위한 선형(Hull)의 Loading 예

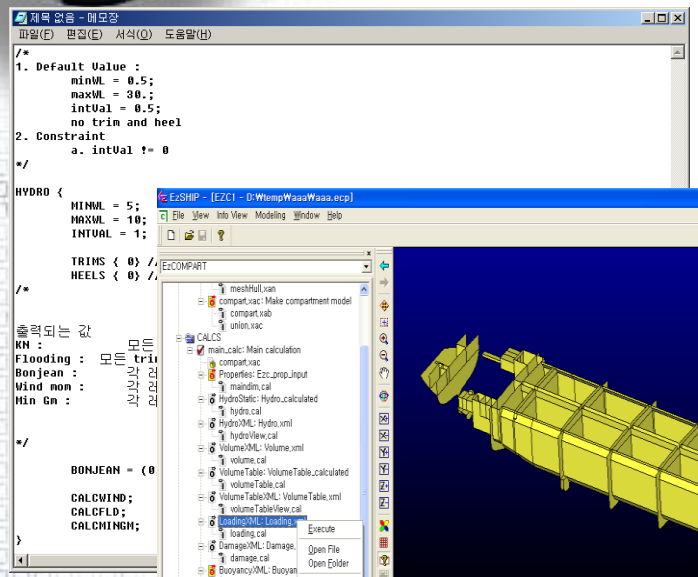
3,700 TEU Container Carrier로 대체

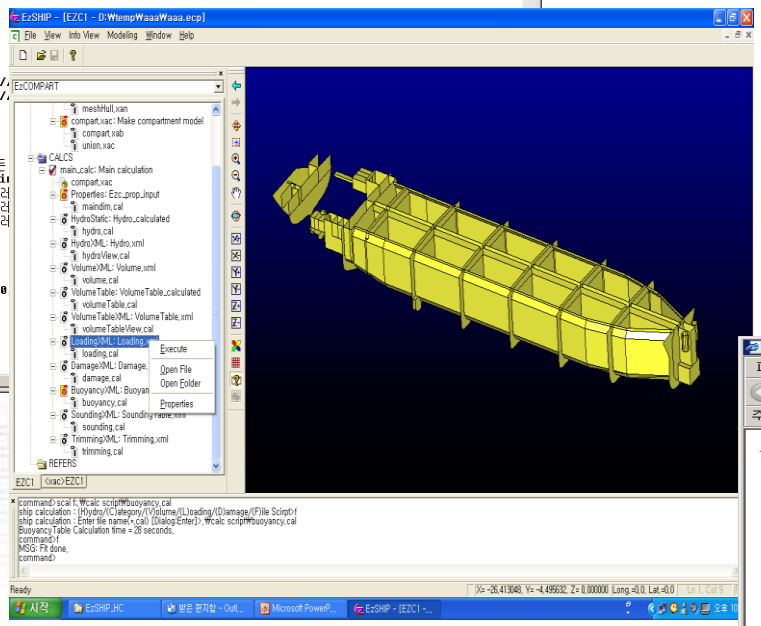


place (hull) @{*};



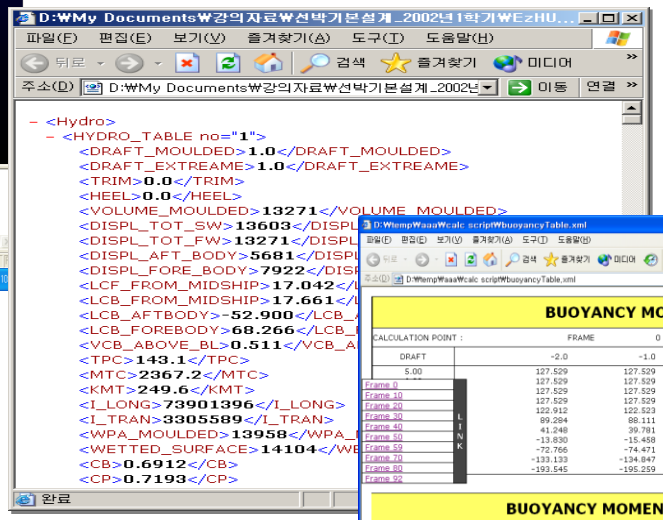
2. 선박 계산
- 선박 계산 수행 절차



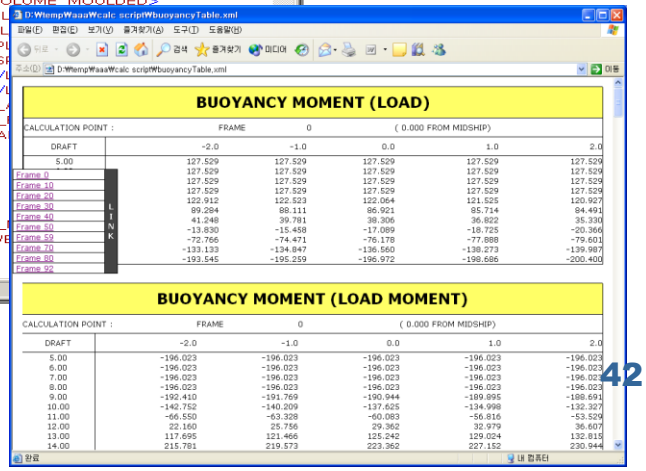


계산용 스크립트 생성
(주요 치수, 흘수 구간, Trim, Heel, 경하중량 분포, 화물 적재 정보)

각종 선박 계산의 순차적인 실행

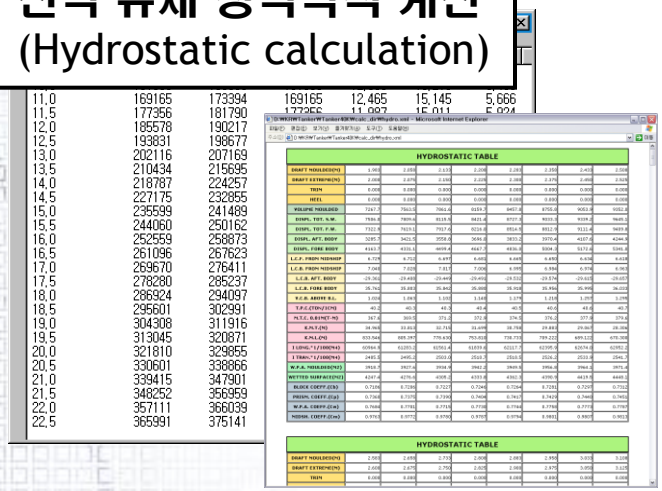


계산된 결과 출력

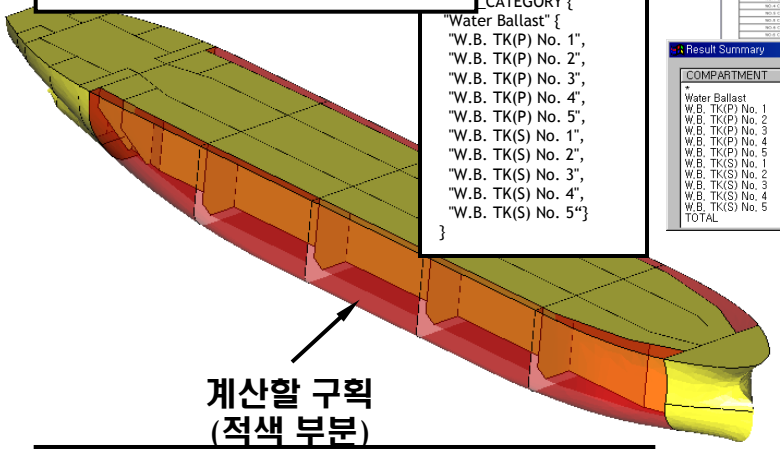


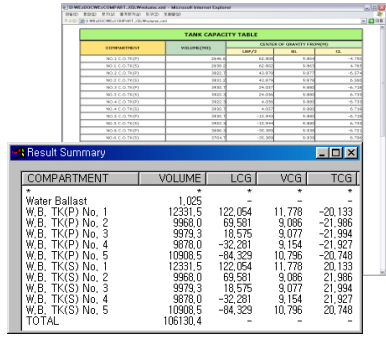
2. 선박 계산
- 선박 계산의 내용

선박 유체 정역학적 계산
(Hydrostatic calculation)



구획 용적 계산
(Volume calculation)

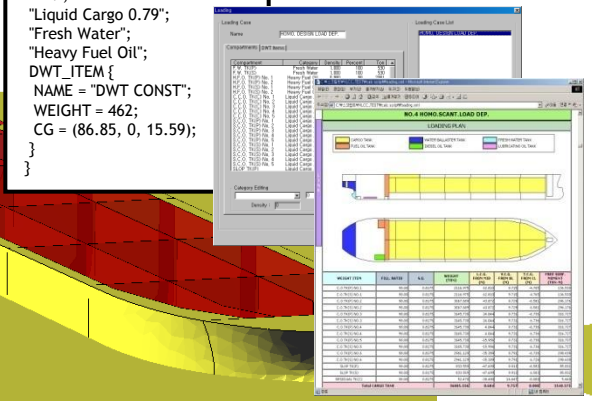




계산할 구획
(적색 부분)

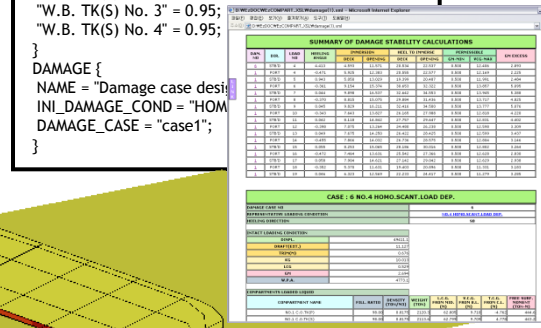
비손상 시 복원 성능 계산
(Intact stability calculation)

평형 상태의
수선면

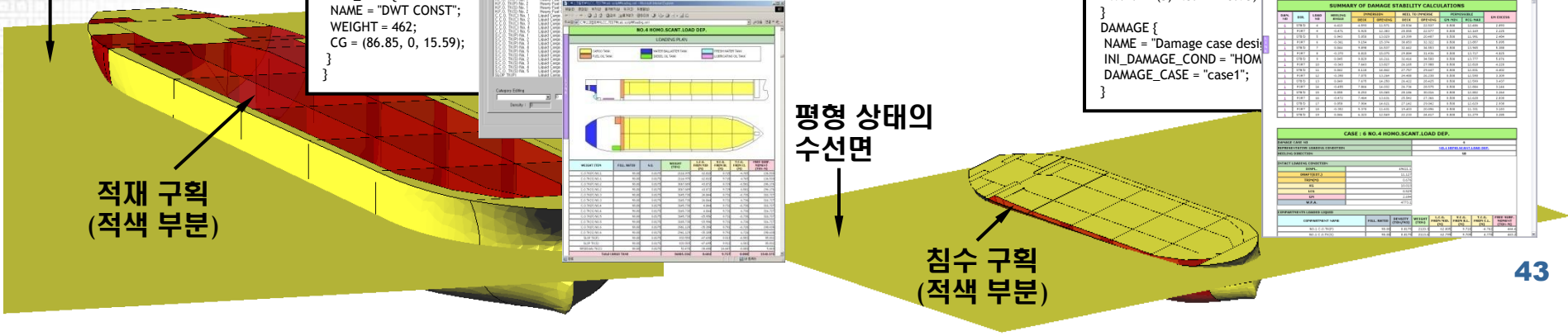


적재 구획
(적색 부분)

손상 시 복원 성능 계산
(Damage stability calculation)



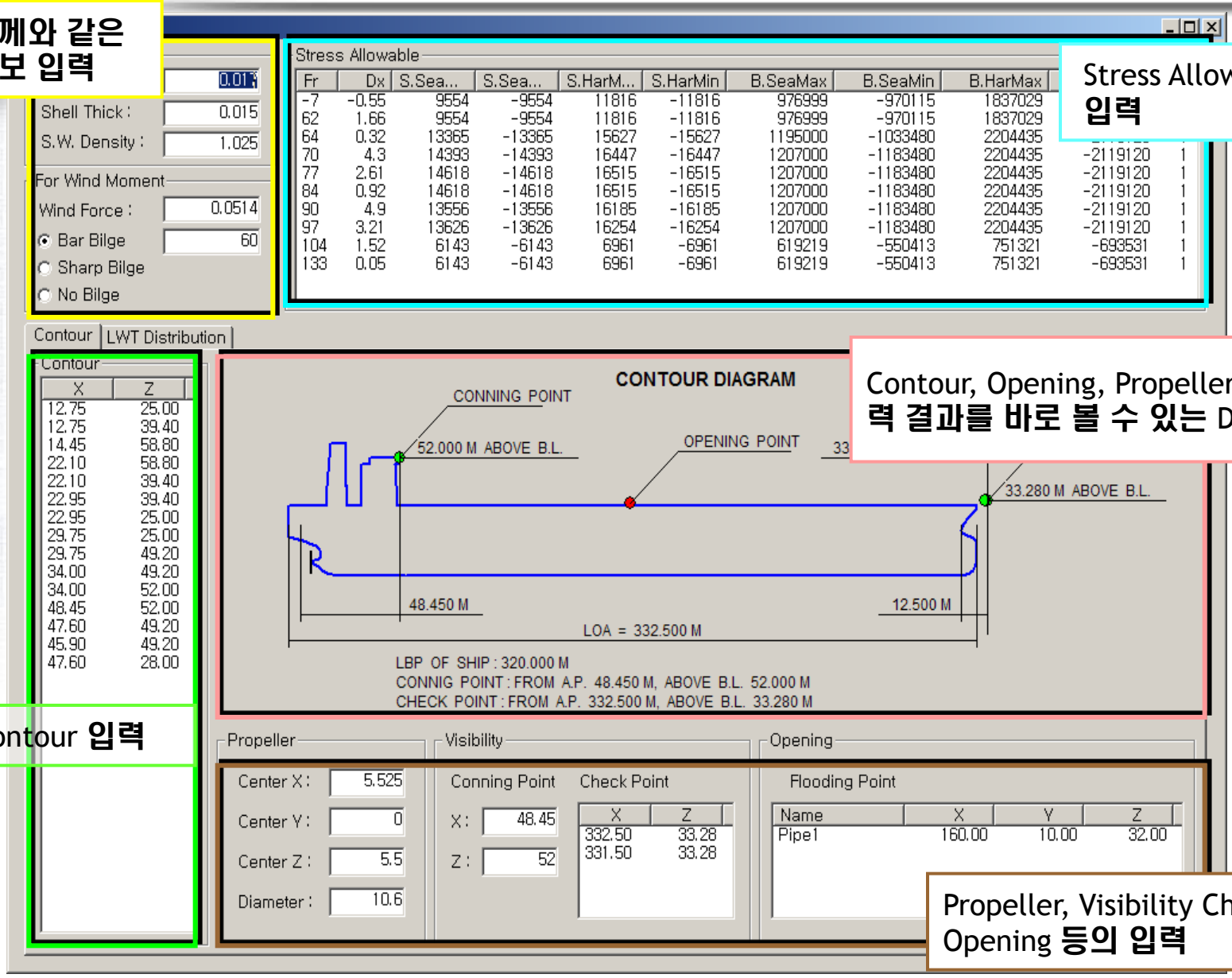
평형 상태의
수선면



침수 구획
(적색 부분)

2. 선박 계산
- 선박 계산 속성 입력 GUI(1)

외판 두께와 같은 기본 정보 입력



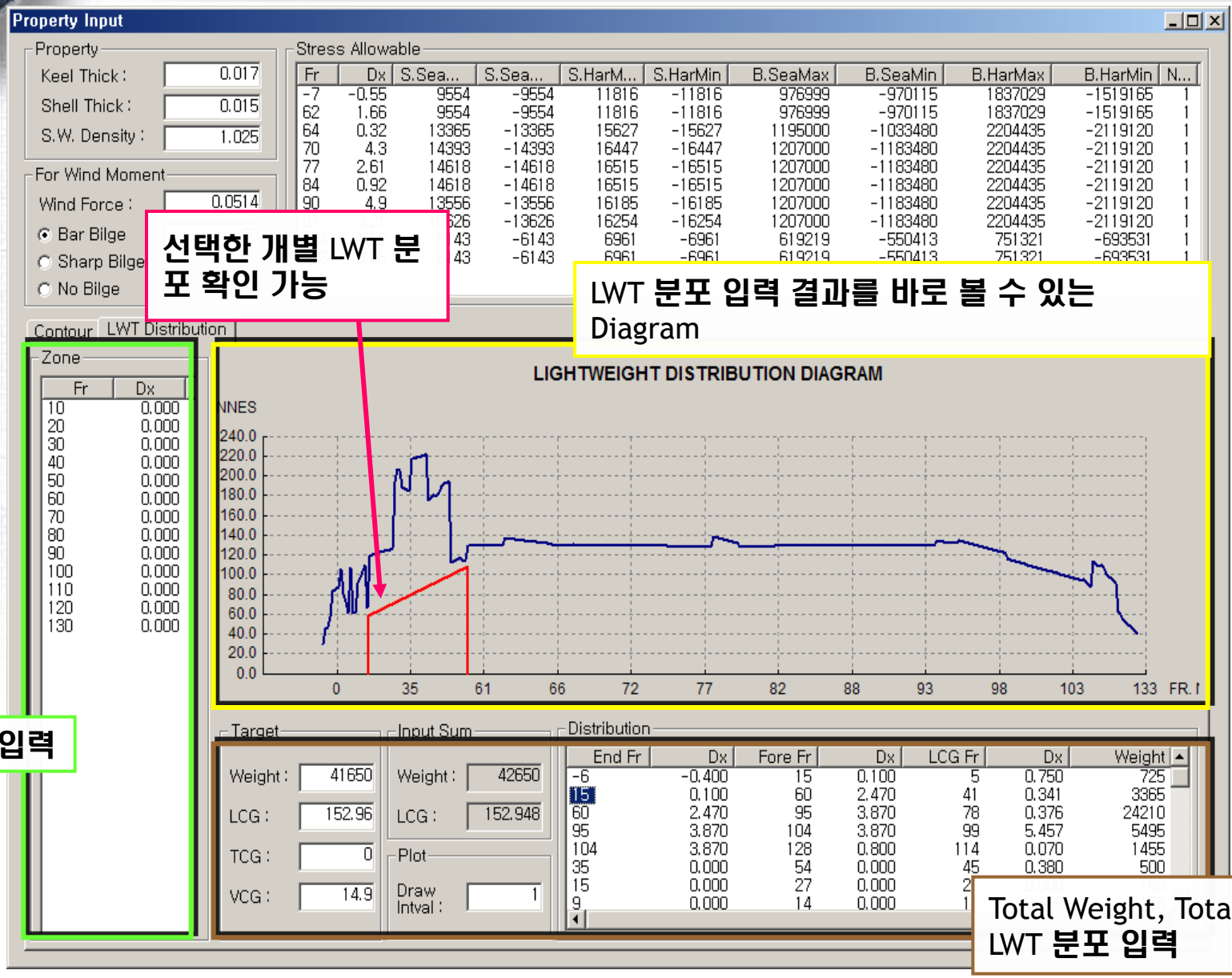
Stress Allowance 입력

Contour, Opening, Propeller 등의 입력 결과를 바로 볼 수 있는 Diagram

Deck Contour 입력

Propeller, Visibility Check, Opening 등의 입력

2. 선박 계산
- 선박 계산 속성 입력 GUI(2)



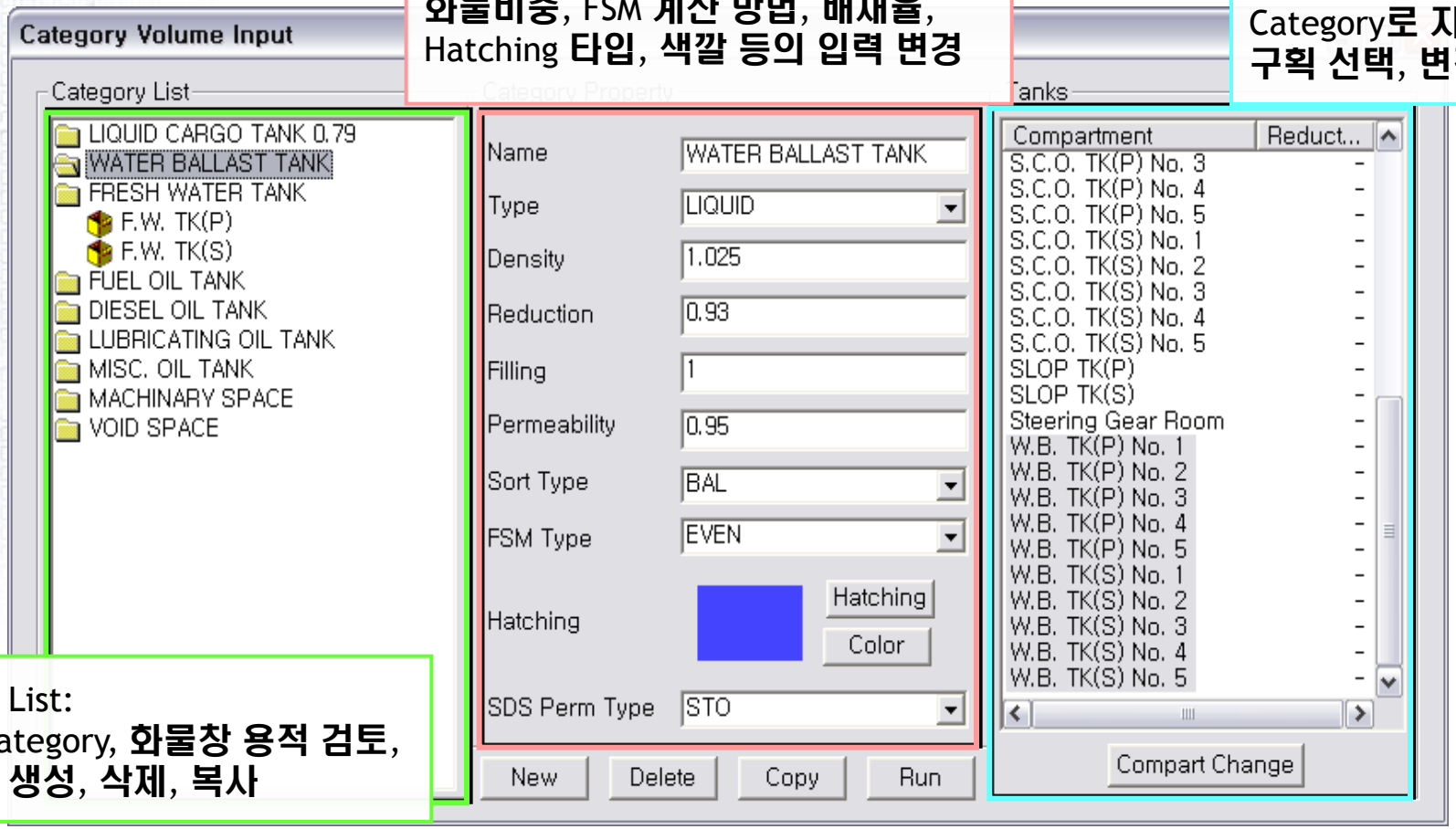
ZONE 입력

Total Weight, Total CG, LWT 분포 입력

2. 선박 계산
- 선박 계산 속성 입력 GUI(3)

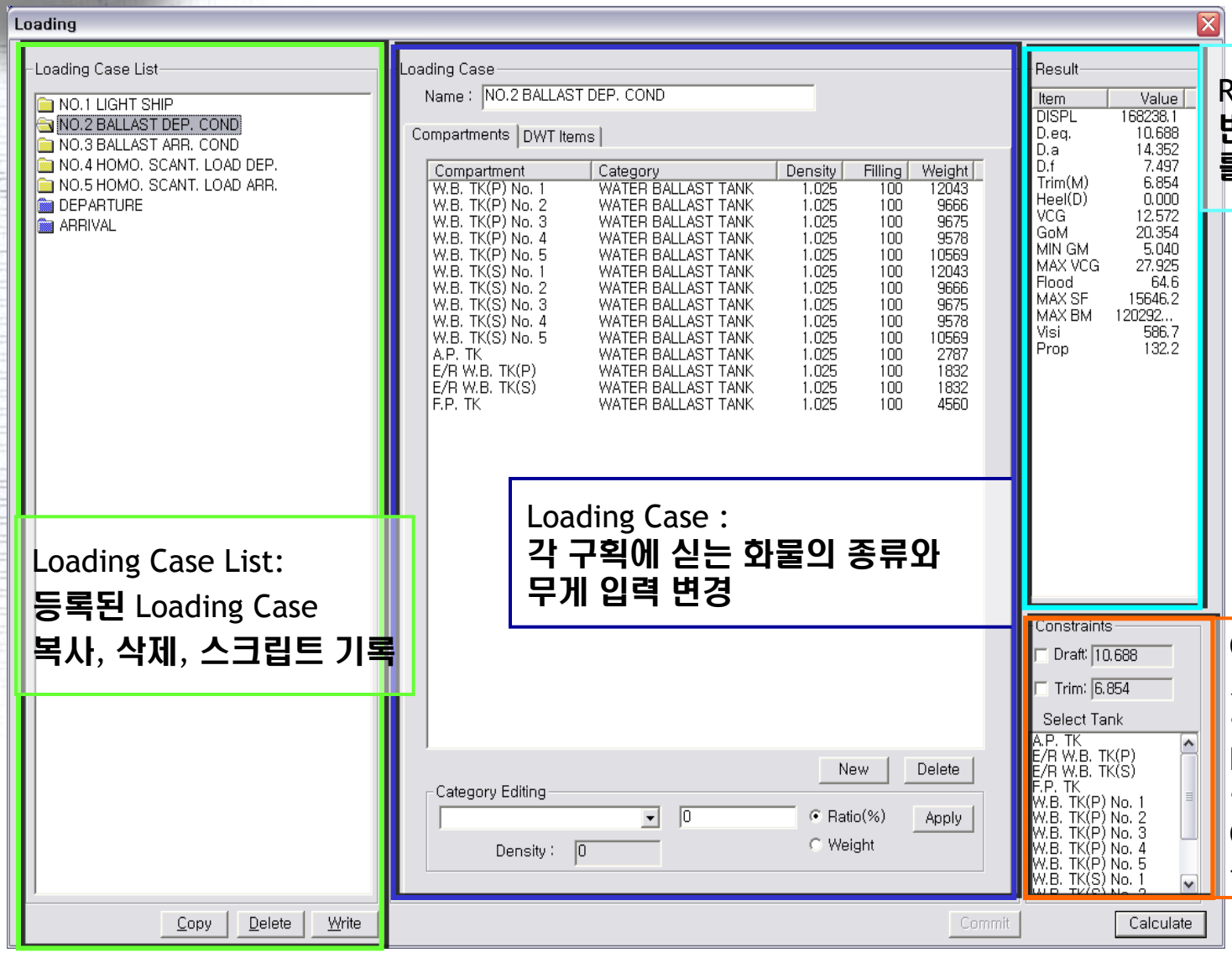
Category Property :
화물비중, FSM **계산 방법**, **배재율**, Hatching **타입**, **색깔 등의 입력 변경**

Tank List:
Category**로 지정할 구획 선택**, **변경**



Category List:
등록된 Category, **화물창 용적 검토**, Category **생성**, **삭제**, **복사**

2. 선박 계산
- 선박 계산 속성 입력 GUI(4)



Loading Case List:
등록된 Loading Case
복사, 삭제, 스크립트 기록

Loading Case :
각 구획에 싣는 화물의 종류와 무게 입력 변경

Result:
변경된 Loading **결과를 바로 확인**

Constraints:
지정한 Tank**의** Filling **을 자동 변경하여,** Draft, Trim **제한 조건을 만족하는** Loading Case
생성

2. 선박 계산
- 산박 계산을 위한 입력 파일 작성(1)

- 선박 계산을 위한 입력 파일 (*.cal)작성
  - 선박 계산을 위한 입력 파일에 들어 가는 정보는 크게 다음과 같은 4가지임
    - 선박 유체 정역학적 계산을 위한 정보
    - ➡ 선박의 비손상 상태에 대한 다양한 값 계산(용적, 침수표면적, 길이/폭/높이 방향 무게 중심, …)
    - 구획 용적 계산에 관련된 정보
    - ➡ 해당 구획에 얼마만큼의 화물을 실을 수 있나를 계산
    - Loading Condition에 관련된 정보
    - ➡ 선박의 구획에 어떤 화물들을 싣고 운항할 것이며 그때 선박의 자세는 어떠한가
    - Damage Stability 계산에 관련된 정보
    - ➡ 선박이 손상을 입었을 때 그 때의 자세는 어떠하며 각종 국제 규약의 조건을 만족하는지 평가

2. 선박 계산
- 산박 계산을 위한 입력 파일 작성(2)

- 선박 유체 정역학적 계산을 위한 정보 입력("hydro.cal")
  - "MAINDM" 블록에서…
    - KEELTHICK: Keel Plate의 두께, Extreme Draft 출력시 이용됨
    - LONGI = (a, b, c): 각 Loading Condition에서 Bending Moment와 Shear Force가 계산될 길이 방향 위치 설정, a와 c사이를 b개로 분할하여 그 위치에서 값을 계산함
  - "LIGHTWT" 블록에서…
    - 이 블록을 입력하지 않으면 각 Loading Condition에 대한 Bending Moment와 Shear Force가 계산 안됨
  - "HYDRO" 블록에서…
    - MINWL: 선박 유체 정역학적 계산이 수행되는 최소 흘수
    - MAXWL: 선박 유체 정역학적 계산이 수행되는 최대 흘수
    - INTVAL: MINWL과 MAXWL을 얼마만큼 분할하여 그 위치에서 선박 유체 정역학적 계산을 수행할 것인가를 결정
    - TRIMS, HEELS: Even Keel 상태뿐만 아니라 Trim, Heeling 상태에서의 선박 유체 정역학적 계산을 수행하고자 할 때(+: 선미 트림, 우현 경사)

2. 선박 계산
- 산박 계산을 위한 입력 파일 작성(3)

- **구획 용적 계산에 관련된 정보 입력**(“volume.cal”)
  - “DEF_CATEGORY” **블록에서…**
    - CATEGORYTYPE: **구획 내에 적재되는 화물의 종류 결정**
    - DENSITY: **화물의 밀도**
    - REDUCTION: **구획 내의 구조 부재**(Stiffener **등**)**로 인한 고려하여 실제 구획 내에 적재 가능한 용적의 비율을 나타냄**
    - FILLING: **그 구획 내 화물을 얼마나 적재할 것인가를 나타냄, 실제 구획 내에 적재되는 화물의 용적은** (**구획의 용적**)*(REDUCTION)*(FILLING)**가 됨**
    - PERMEABILITY: **구획이 손상되었을 때 해수에 의해 얼마나 침수될 것인가를 나타냄**

2. 선박 계산
- 산박 계산을 위한 입력 파일 작성(4)

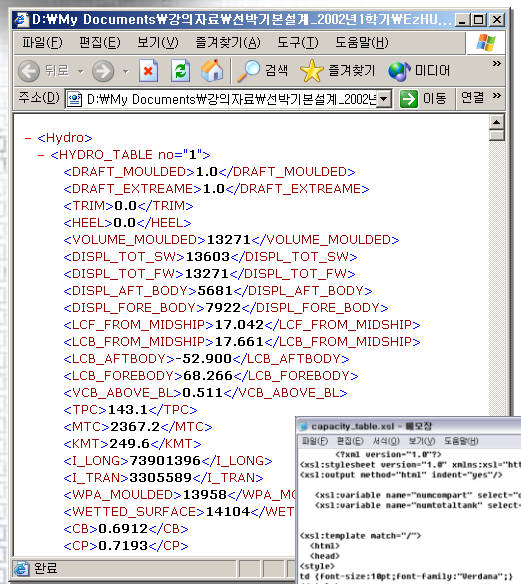
- Loading Condition에 관련된 정보 입력(“loading.cal”)
  - “LOADING_CASE” 블록에서…
    - WAIT: 해당 Loading Case에 대한 계산을 바로 수행하지 않고 사용자의 최종 입력을 기다리라는 키워드
    - DWT_ITEM: 컨테이너와 같이 특정 위치에 재화 중량을 분포시키기 위한 키워드, 컨테이너는 베이 단위로 분포시킴
    - TRIMTOL, HEELTOL: 각 Loading Condition에 대한 선박 자세의 반복 계산시 이용되는 Tolerance 값
    - BASE_LOADING_CASE=“name”: 앞서 정의된 Loading Case의 값의 일부를 가져오고자 할 때 이용되는 키워드

2. 선박 계산
- 산박 계산을 위한 입력 파일 작성(5)

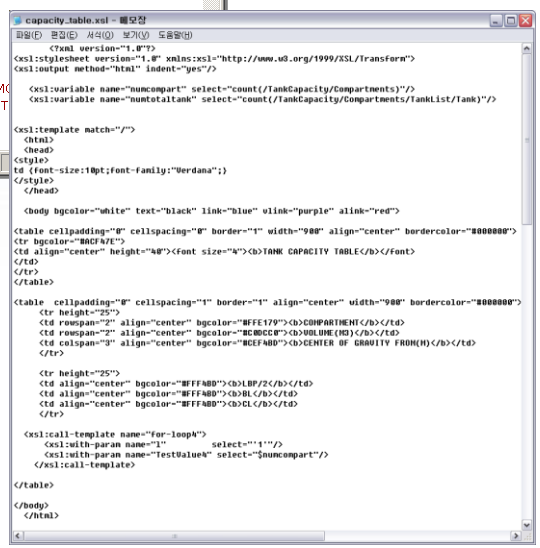
- Damage Stability 계산에 관련된 정보 입력("damage.cal")
  - "CRITERIA" 블록에서…
    - LINE="name": 선박 침수의 기준이 되는 Deck Line 등을 정의하기 위한 키워드
    - OPENING="name": 선박 침수의 기준이 되는 Deck 상의 Pipe와 같은 것을 정의하기 위한 키워드
    - MARGIN("rule name", "name"): Margin Line의 모든 Opening과 일반 Opening 중에서 non-tight type의 점이 수선면 위에 있어야 함. 입력되는 점들이 하나도 없을 때는 이 조건을 만족한 것으로 함
    - HEELING("rule name", "name", a, b): 횡경사 각도가 지정 각도보다 작아야 함. b는 MARGIN 만족시 a보다 기준이 낮아진 각도임.
    - RANGE_GZ("rule name", "name", a): 평형 위치에서부터 a 각도를 더한 위치에서의 GZ 값을 구함. 이 값과 그 사이의 모든 GZ 값이 양이면 조건을 만족한다고 판단함
    - MAX_GZ("rule name", "name", a, b): 우선 Damage시 평형점에 a 각도를 더한 구간에서의 최대 GZ 값이 b 이상이 되면 조건을 만족한 것으로 판단함
    - AREA_GZ("rule name", "name", a, b): Heeling 각도 a까지의 양의 GZ 곡선하 면적이 주어진 값 b 보다 크면 조건을 만족한 것으로 판단함
    - MIN_GM("rule name", "name", a): 요구 GM a 값보다 크면 조건을 만족한 것으로 판단함

2. 선박 계산
- 선박 계산의 결과

XML 파일 출력

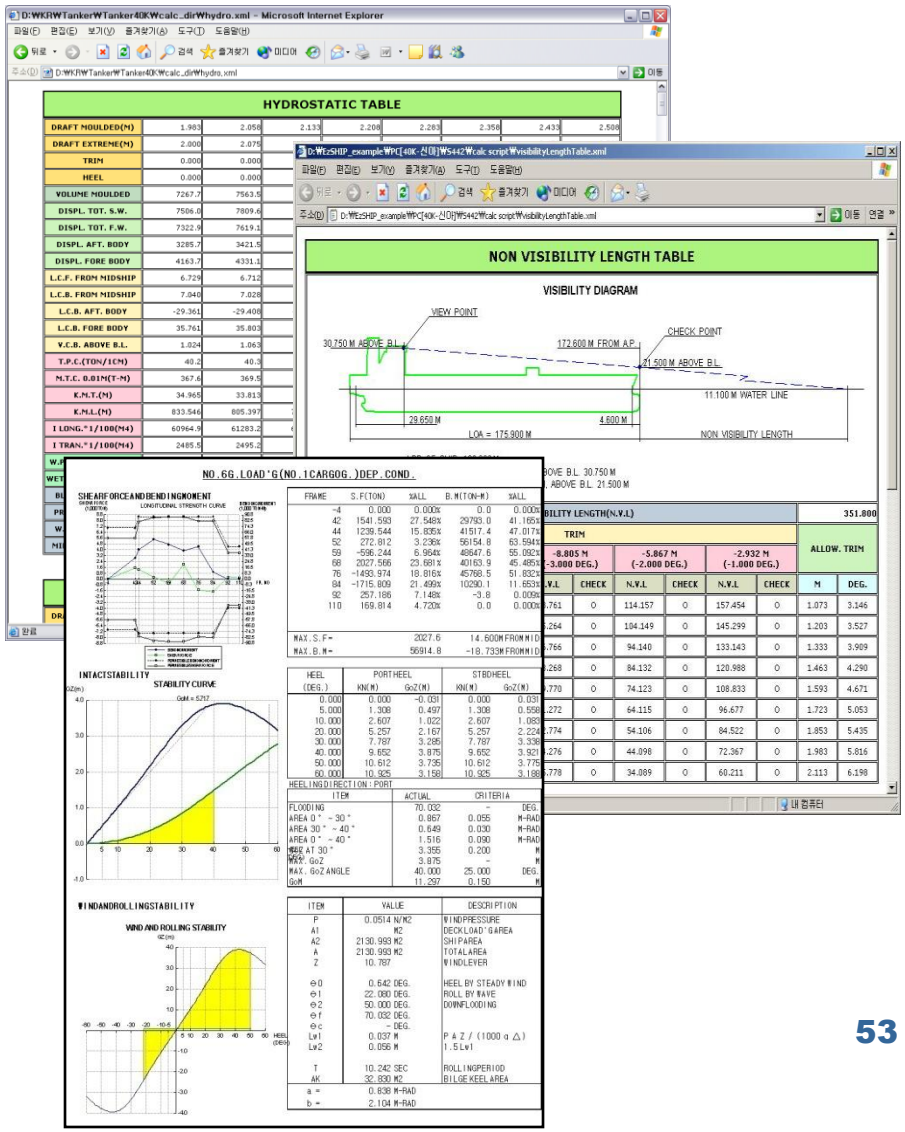


+



XSLT Document
(XML ➡ HTML 변환)

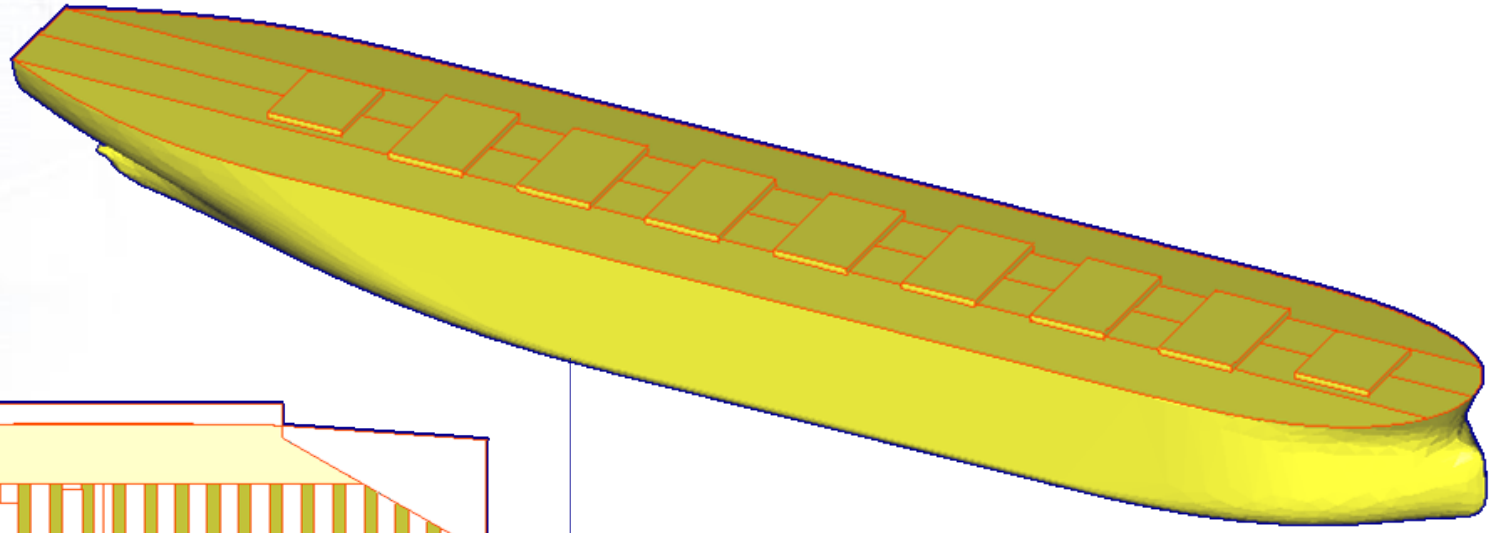
HTML 문서로 표현

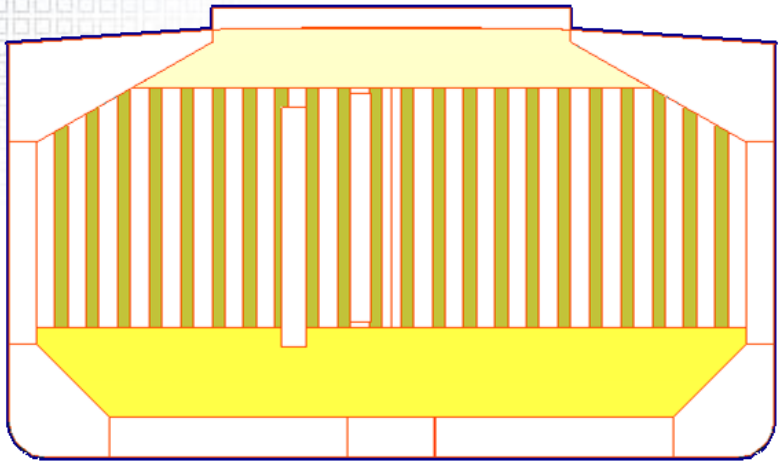


참고 Slides

살물선의 모델링을 위한 Hatch Coaming의 정의 예(1)

UNION 연산을 이용한 Hatch Coaming 정의

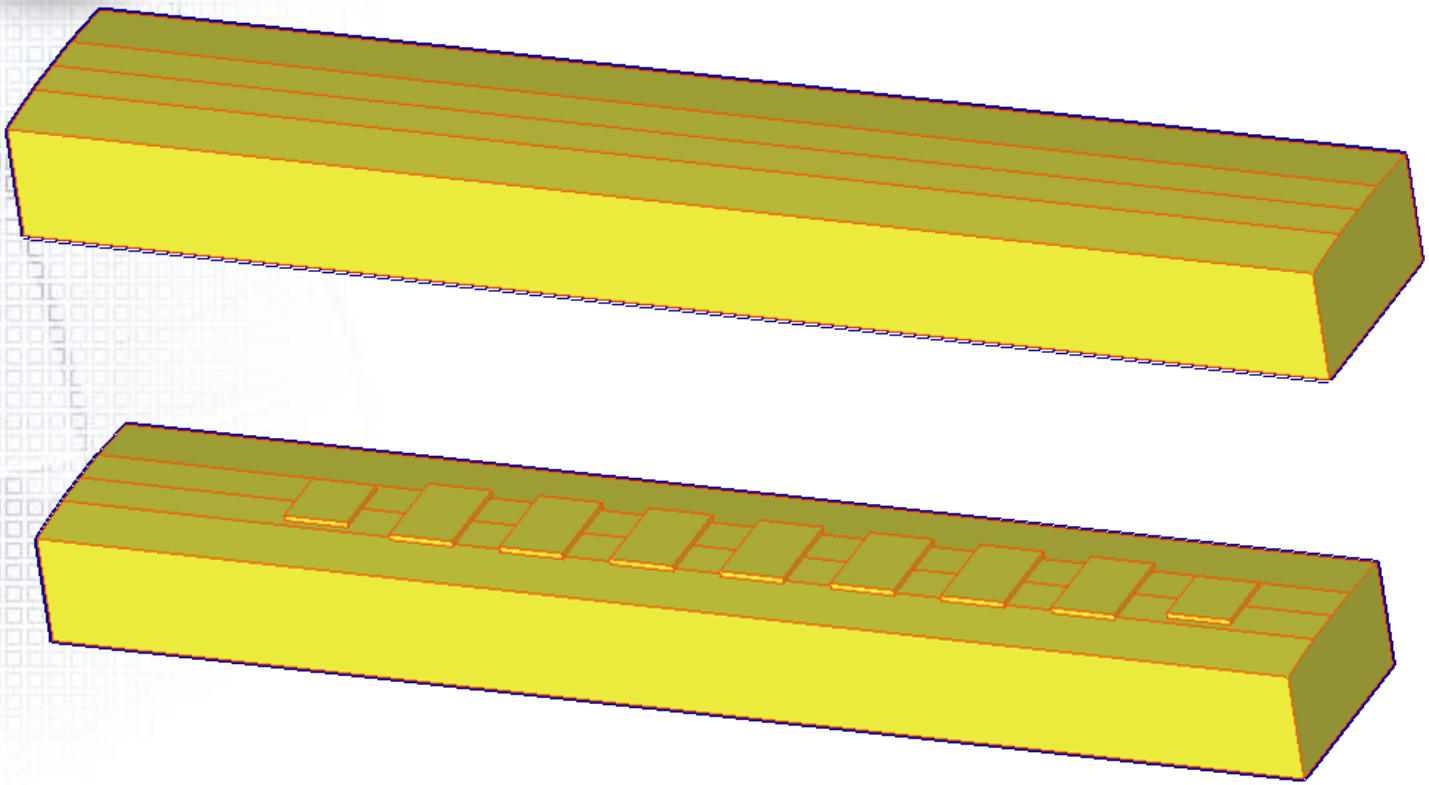




살물선의 모델링을 위한 Hatch Coaming의 정의 예(2)

UNION 연산

참고: Base Box 외부에 Cutter 위치 시 기존 모델과 Union이 됨



Deck 생성

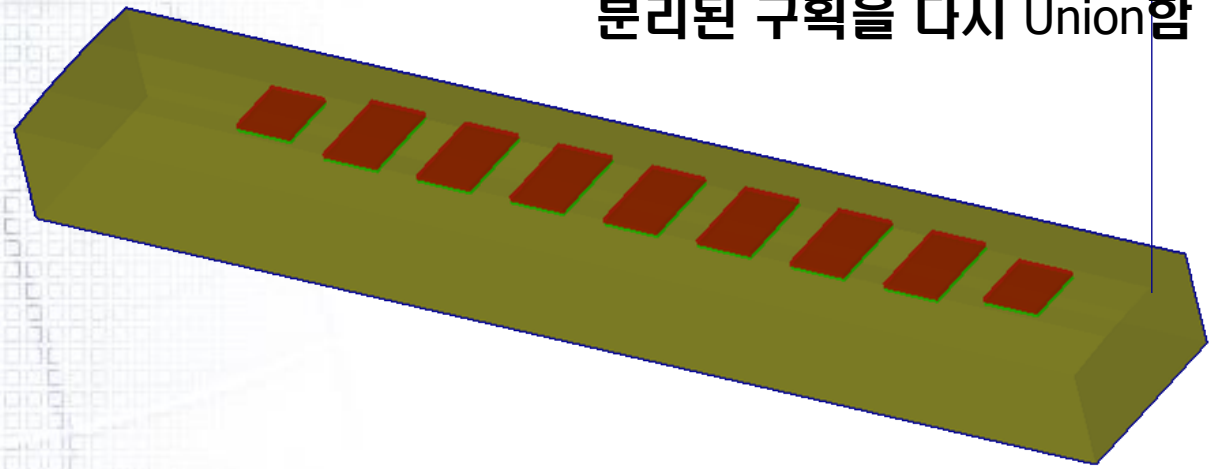
Hatch Box 정의 후 Union 연산

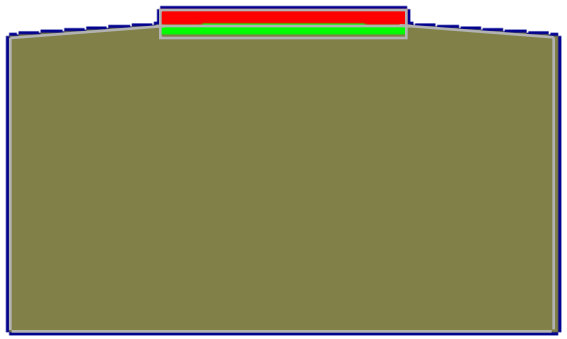
```
box hatch1 { (279#x0, -7.65, $D), (295#x0, 7.65, 26.45)};
box hatch2 { (251#x0, -10.5, $D), (267#x0, 10.5, 26.45)};
box hatch3 { (223#x0, -10.5, $D), (239#x0, 10.5, 26.45)};
box hatch4 { (195#x0, -10.5, $D), (211#x0, 10.5, 26.45)};
union (hatch1, hatch2, hatch3, hatch4, hatch5, hatch6, hatch7, hatch8, hatch9);
```

살물선의 모델링을 위한 Hatch Coaming의 정의 예(3)

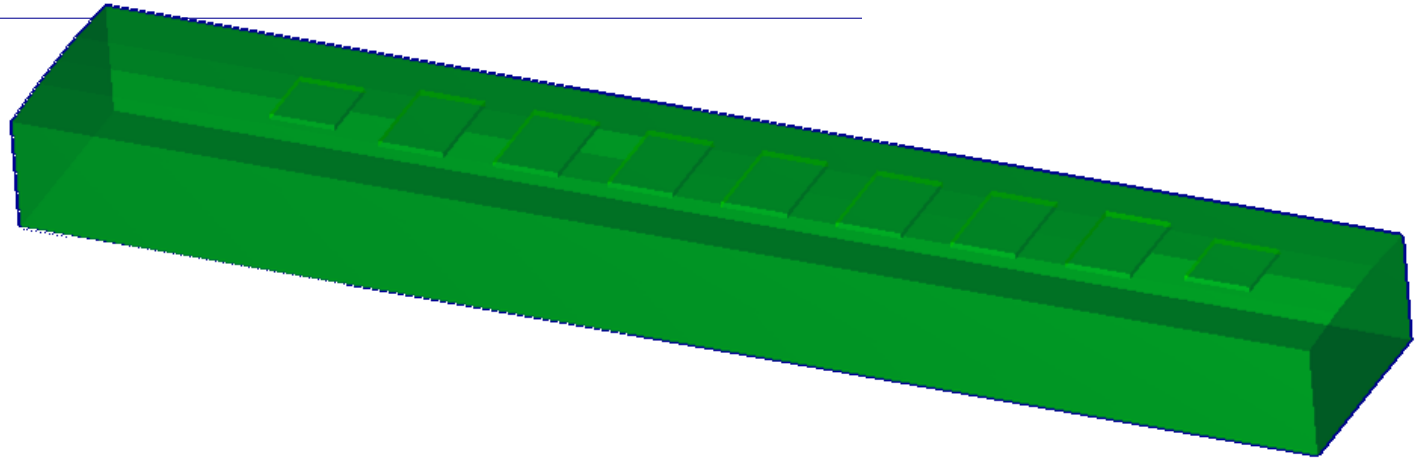
UNION 연산

세 개의 구획으로 분리되며,
분리된 구획을 다시 Union함



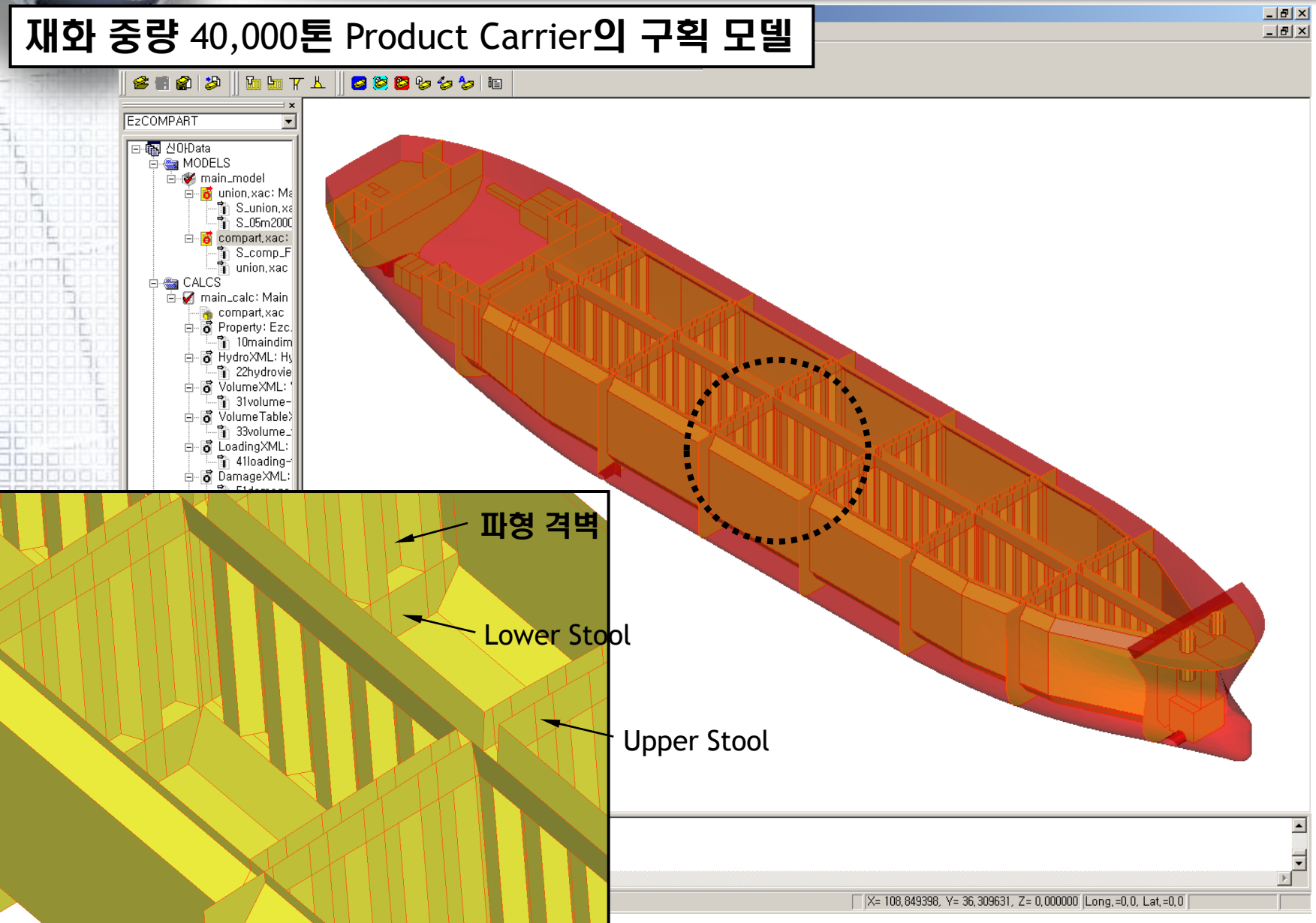


union @{*};

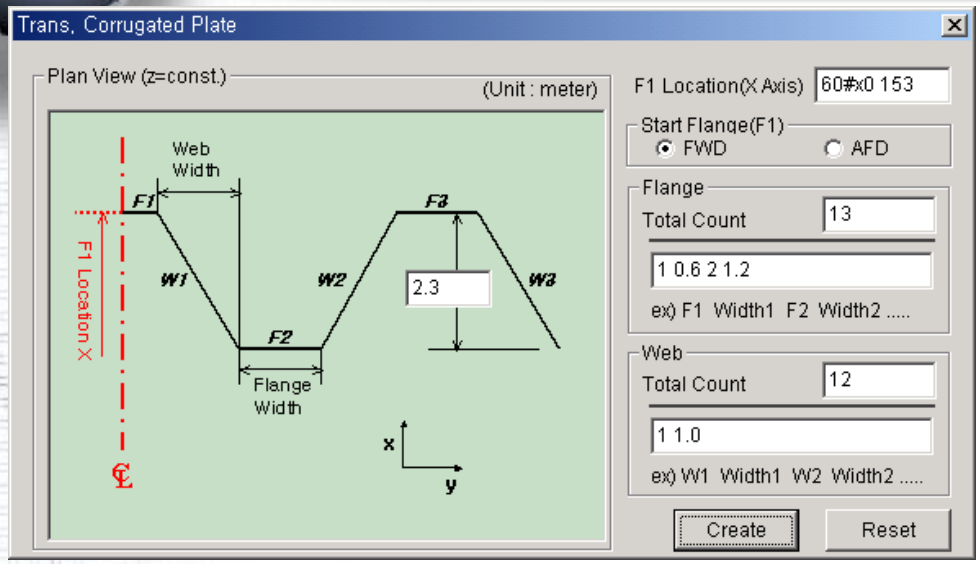


파형 격벽, Lower/Upper Stool의 예

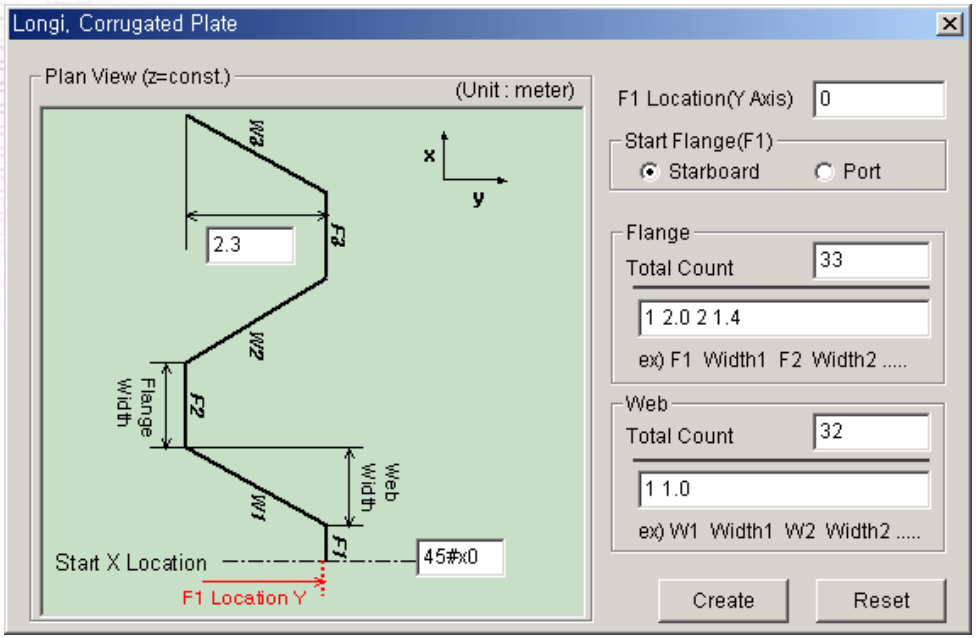
재화 중량 40,000톤 Product Carrier의 구획 모델



파형 격벽(Corrugated Bulkhead)의 정의를 위한 GUI

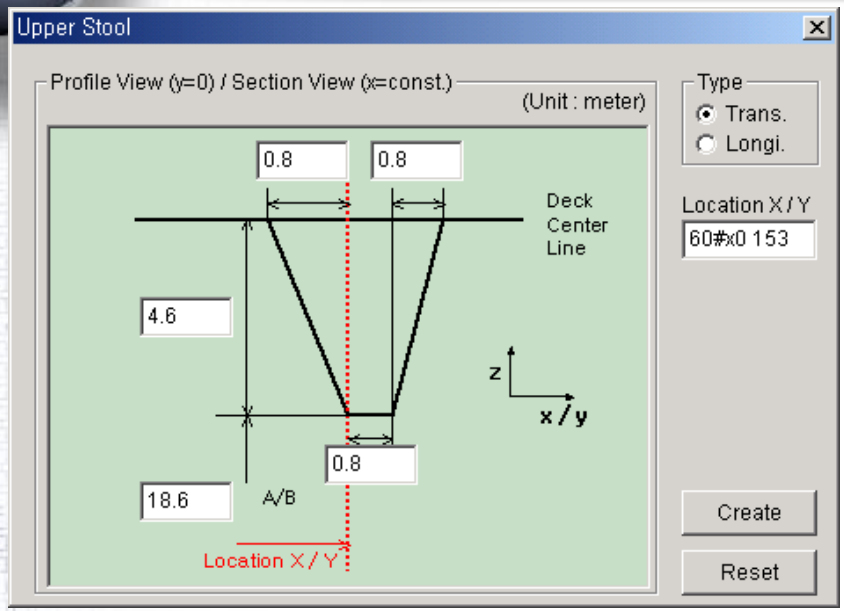


횡 방향 파형 격벽(Corrugated Bulkhead)**을 만드는** polyline, skinsurf **스크립트 생성**

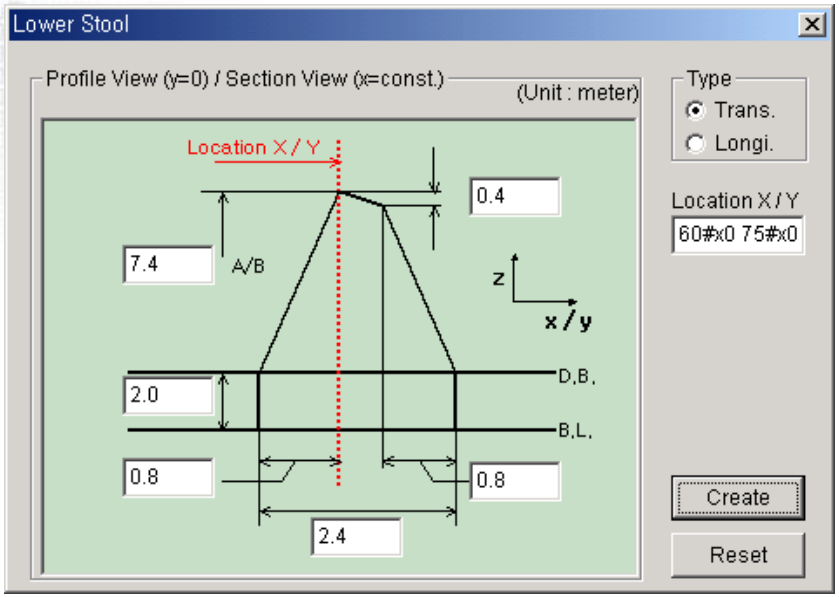


종 방향 파형 격벽을 만드는 polyline, skinsurf **스크립트 생성**

Lower/Upper Stool의 정의를 위한 GUI

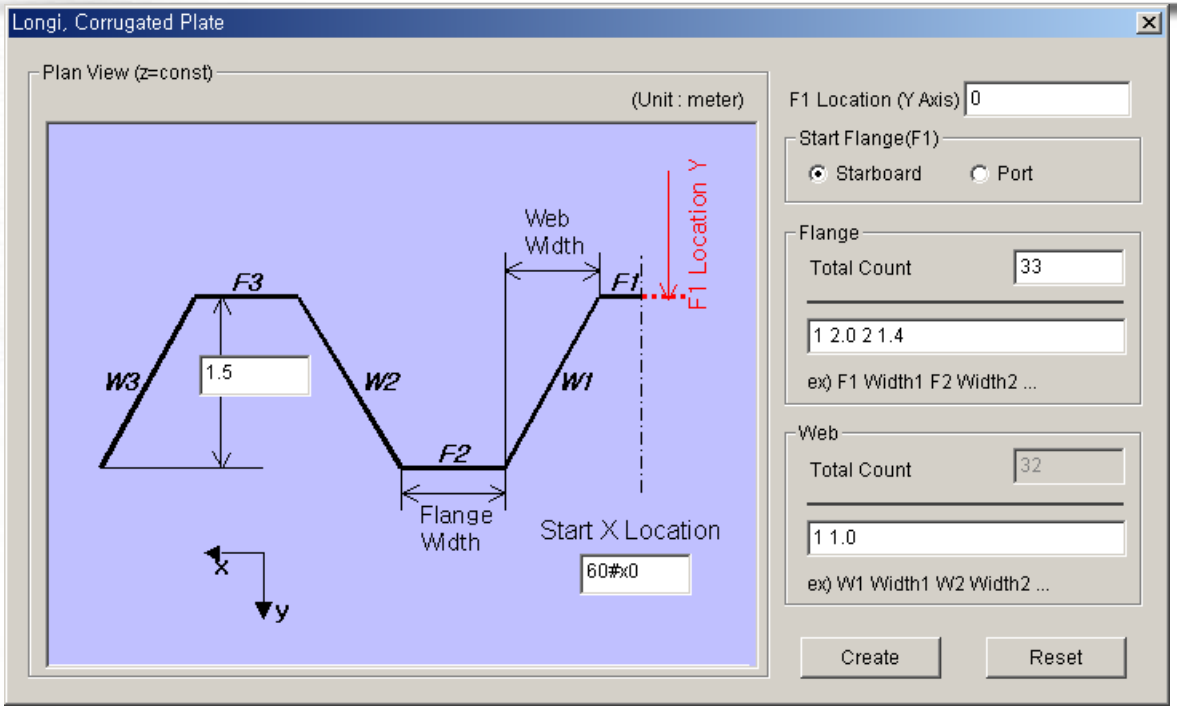


Upper Stool을 구성하는 polyline, skinsurf 스크립트 생성



Lower Stool을 구성하는 polyline, skinsurf 스크립트 생성

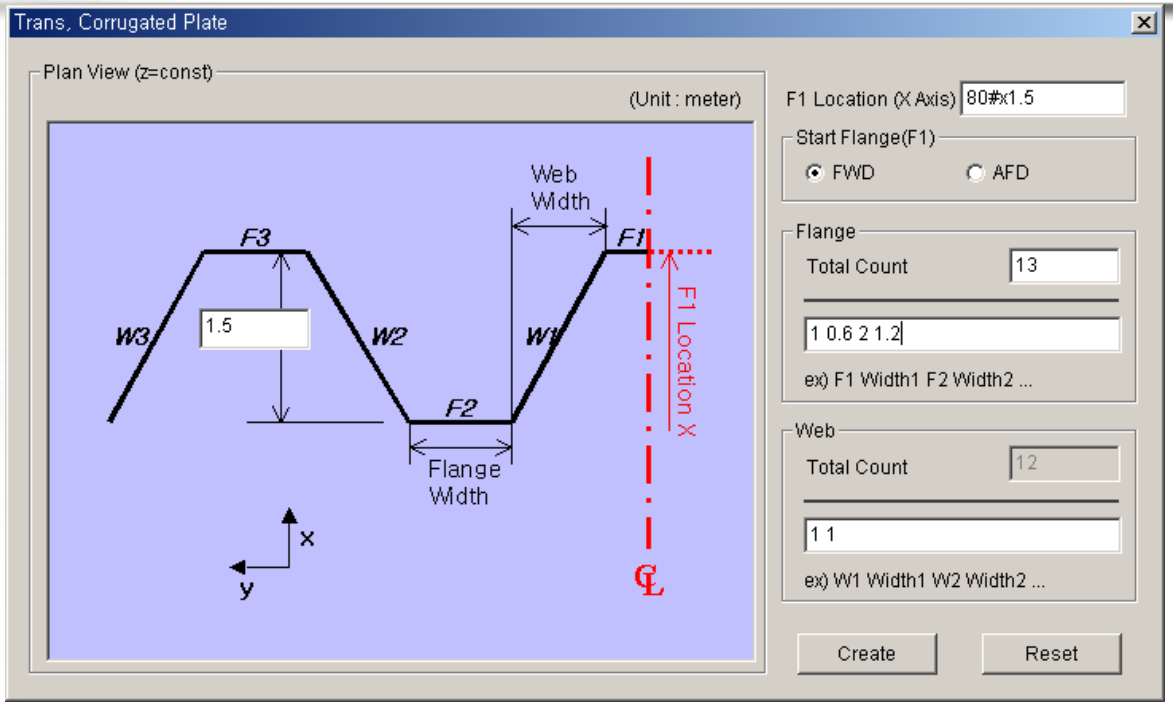
종 방향 파형 격벽의 정의를 위한 GUI



입력 데이터

1. 1.5 m: Corrugated BHD**의** Y **방향 길이** <- Lower **및** Upper Stool**의 폭과 동일해야 함**.
2. Start X Location: **횡격벽과는 달리 종격벽의 경우에는 어디 위치에서 시작하는지를 입력해야 함**.
3. F1 Location(Y Axis): **첫 번째** Flange (F1) **를 기준으로 정의하고자 하는** Corrugated BHD**의** Y**방향에서의 위치**.
4. **첫 번째** Flange(F1)**가** Corrugated BHD**정의를 해나감에 있어 우현방향에 존재하는지, 좌현방향인지 설정**.
5. Total Count: Flange**의 총 개수가 몇 개인지 정의**.
6. **각** Flange**의 폭을 정의. 다만 제일 마지막** Flange**의 경우에는 폭의 크기가 무한대로 생성되며, 화물창길이 등에 의해 결정됨**.
7. Web**도 동일 개념**.
8. Create **버튼: 스크립트를 만들라는 명령이며 대화창이 소멸되진 않는다.** Reset **버튼: 데이터 초기화**.

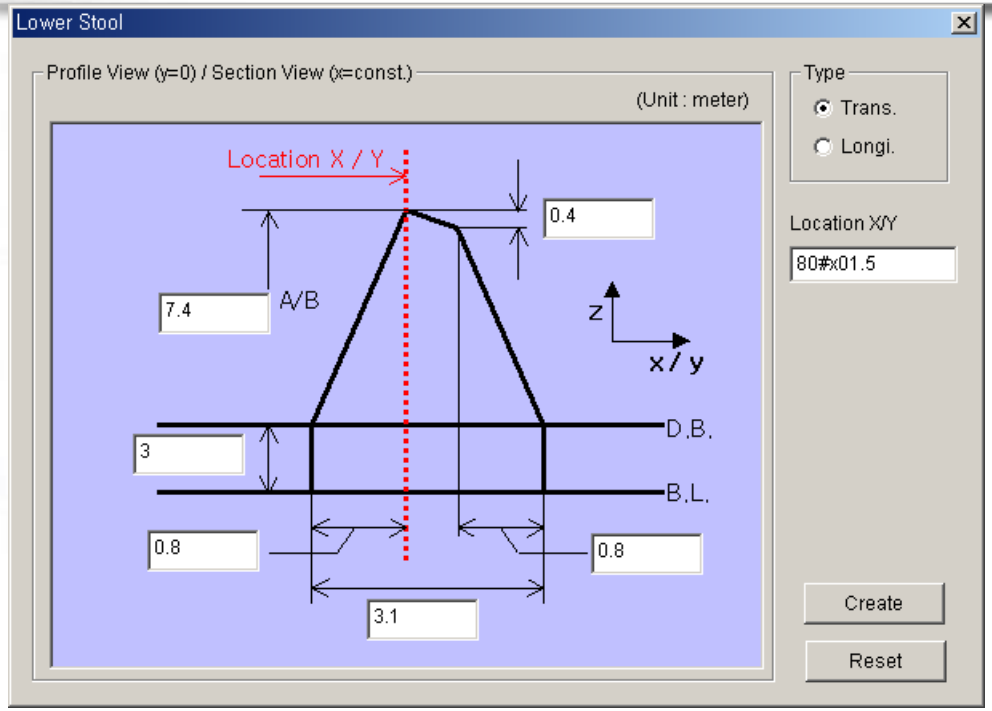
횡 방향 파형 격벽의 정의를 위한 GUI



입력 데이터

1. 1.5 m: Corrugated BHD**의** X **방향 길이** <- Lower **및** Upper Stool**의 폭과 동일해야 함**.
2. F1 Location(X Axis): **첫 번째** Flange (F1) **를 기준으로 정의하고자 하는** Corrugated BHD**의** X**방향에서의 위치. 동일한 형상을 갖는** X**방향의 여러 위치 입력가능**. Frame**형식이나** meter**형식 모두 표현 가능**.
3. **첫 번째** Flange(F1)**가** Corrugated BHD**정의를 해나감에 있어 선수방향에 존재하는지, 선미방향인지 설정**.
4. Total Count: Starboard**에 존재하는** Flange**의 총 개수가 몇 개인지 정의**.
5. **각** Flange**의 폭을 정의. 다만 제일 마지막** Flange**의 경우에는 폭의 크기가 무한대로 생성되며, 배폭 등에 의해 결정됨**. Ex) 1 0.6 5 1.2 : **첫번째** Flange**부터** F4**까지는 모두** 0.6m **폭이고,** F5**이상의 경우에는 모두** 1.2m **폭**.
6. Web**도 동일 개념**.
7. Create **버튼: 스크립트를 만들라는 명령이며 대화창이 소멸되진 않는다**. Reset **버튼: 데이터 초기화**.

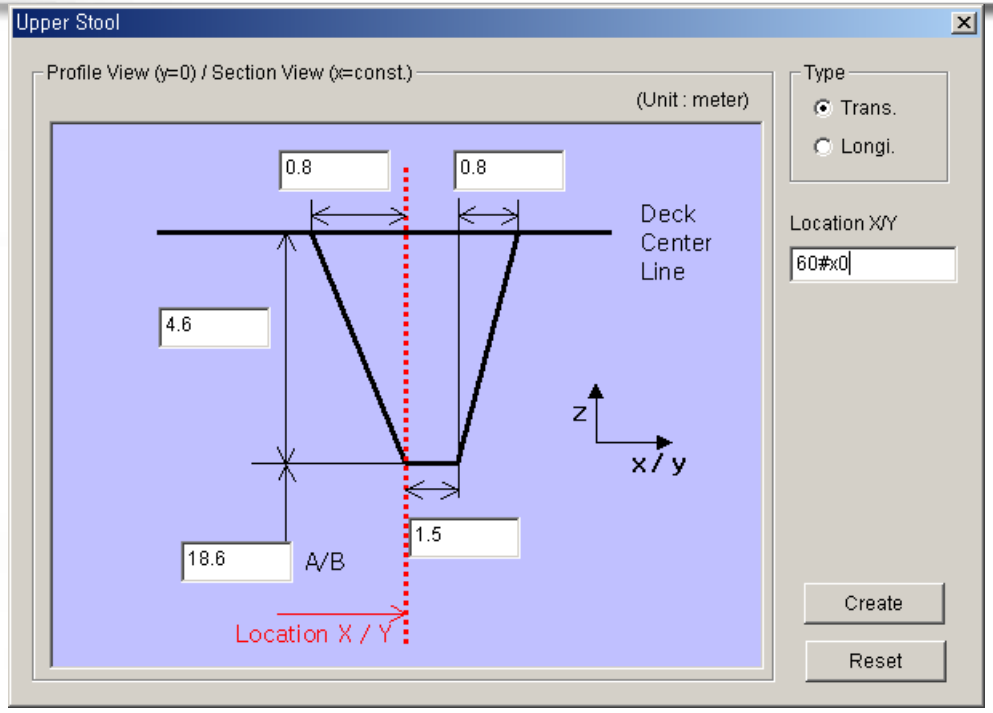
Lower Stool의 정의를 위한 GUI



입력 데이터

1. **대화창 내의 그림상에서의 입력데이터 : 직접 이해가 되리라 예상됨.**
2. Type(Tans., Longi.): **횡격벽을 위한 Stool인지, 종격벽을 위한 Stool인지 구분.**
3. Location X / Y: **정의하는 Stool이 존재하는 위치를 기준선에 대한 좌표로 입력. 횡격벽용 Stool이라면 X위치를 입력하면 되며,** Frame**방식의 입력도 지원함. 또한** Multi**데이터 입력가능.**
4. Create **버튼: 스크립트를 만들라는 명령이며 대화창이 소멸되진 않는다.** Reset **버튼: 데이터 초기화.**

Upper Stool의 정의를 위한 GUI



입력 데이터

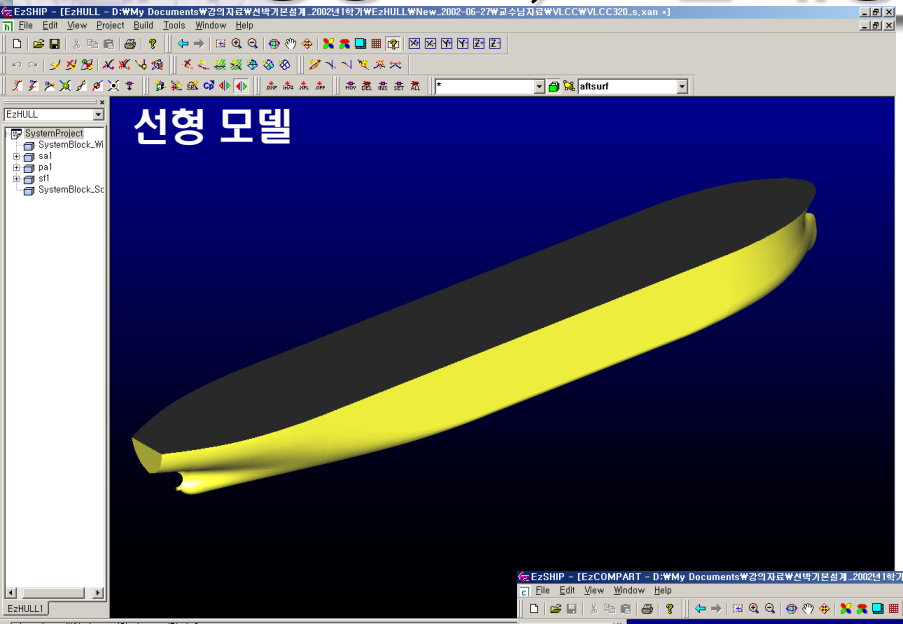
1. **대화창 내의 그림상에서의 입력데이터 : 직접 이해가 되리라 예상됨.**
2. Type(Tans., Longi.): **횡격벽을 위한 Stool인지, 종격벽을 위한 Stool인지 구분.**
3. Location X / Y: **정의하는 Stool이 존재하는 위치를 기준선에 대한 좌표로 입력. 횡격벽용 Stool이라면 X위치를 입력하면 되며, Frame방식의 입력도 지원함. 또한 Multi데이터 입력가능. Multi데이터 입력가능.**
4. Create **버튼: 스크립트를 만들라는 명령이며 대화창이 소멸되진 않는다.** Reset **버튼: 데이터 초기화.**

4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 구획 모델(1)

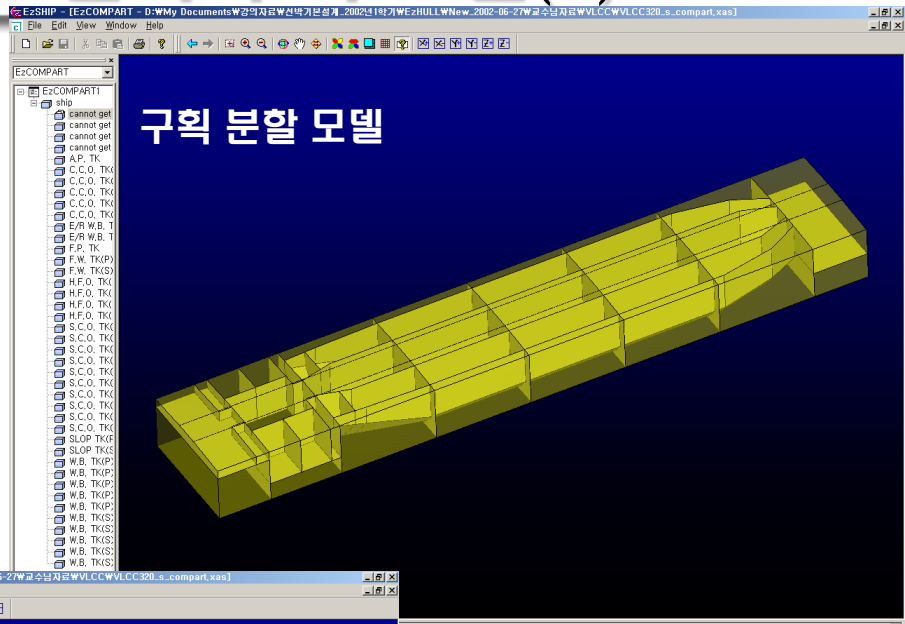
Principal Dimensions

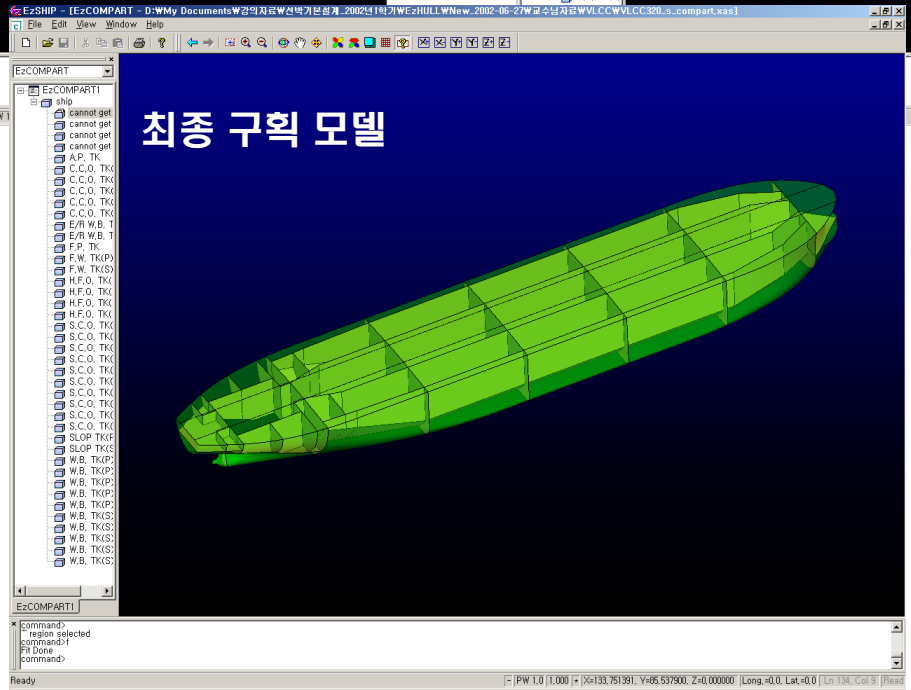
항목		값	비고
주요제원	LOA	332.0 m	
	LBP	320.0 m	
	B	60.0 m	
	D	30.5 m	
	Td / Ts	21.0 / 22.5 m	
Cargo Capacity		320,000 MT	at Ts
Speed		16 knots	at Td
Main Engine	Type	SULZER 7RTA84T-D	
	MCR	39,060 PS x 76.0 rpm	
	NCR	35,150 PS x 73.4 rpm	
Propeller Diameter		10.2 m	

4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 구획 모델(2)

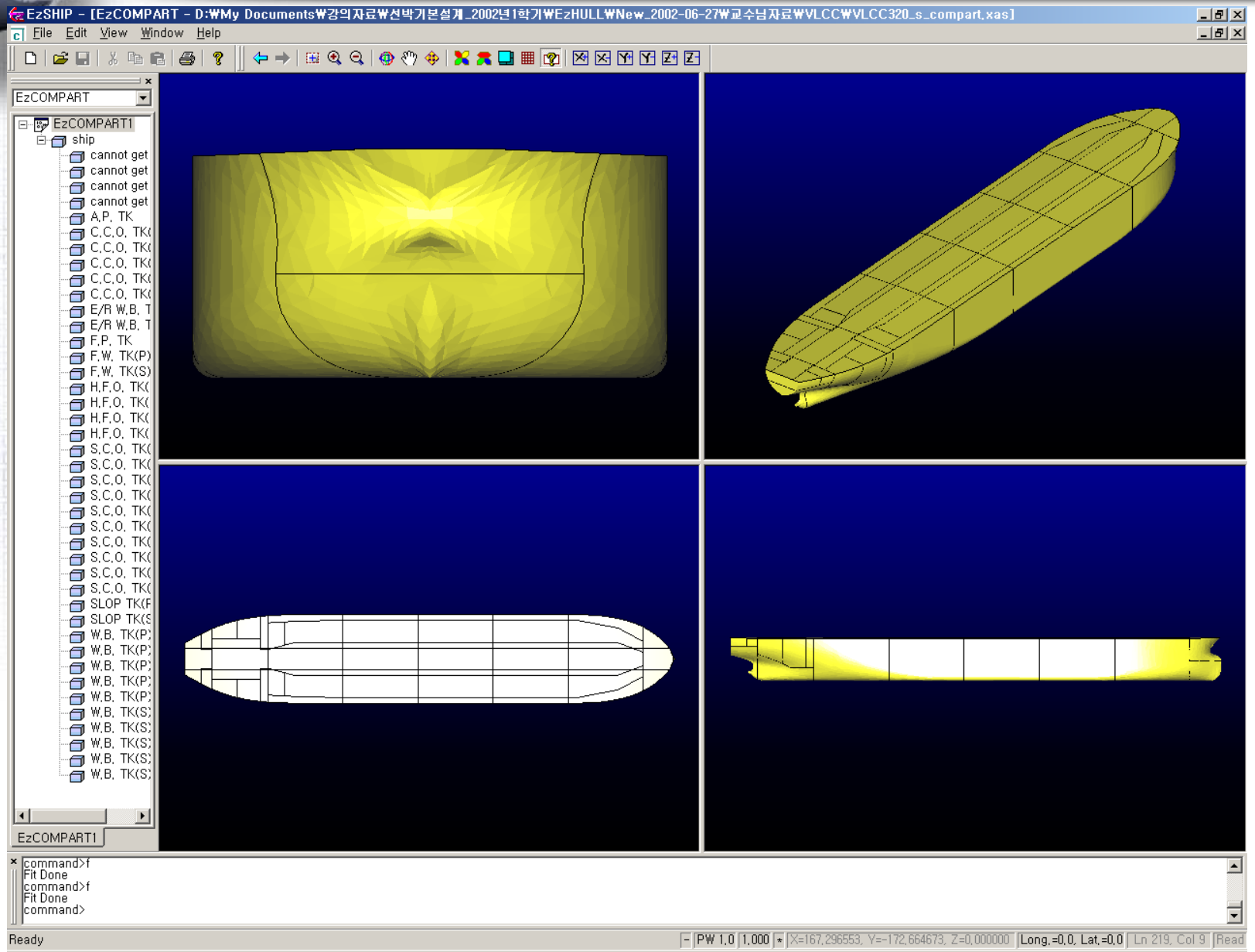


+

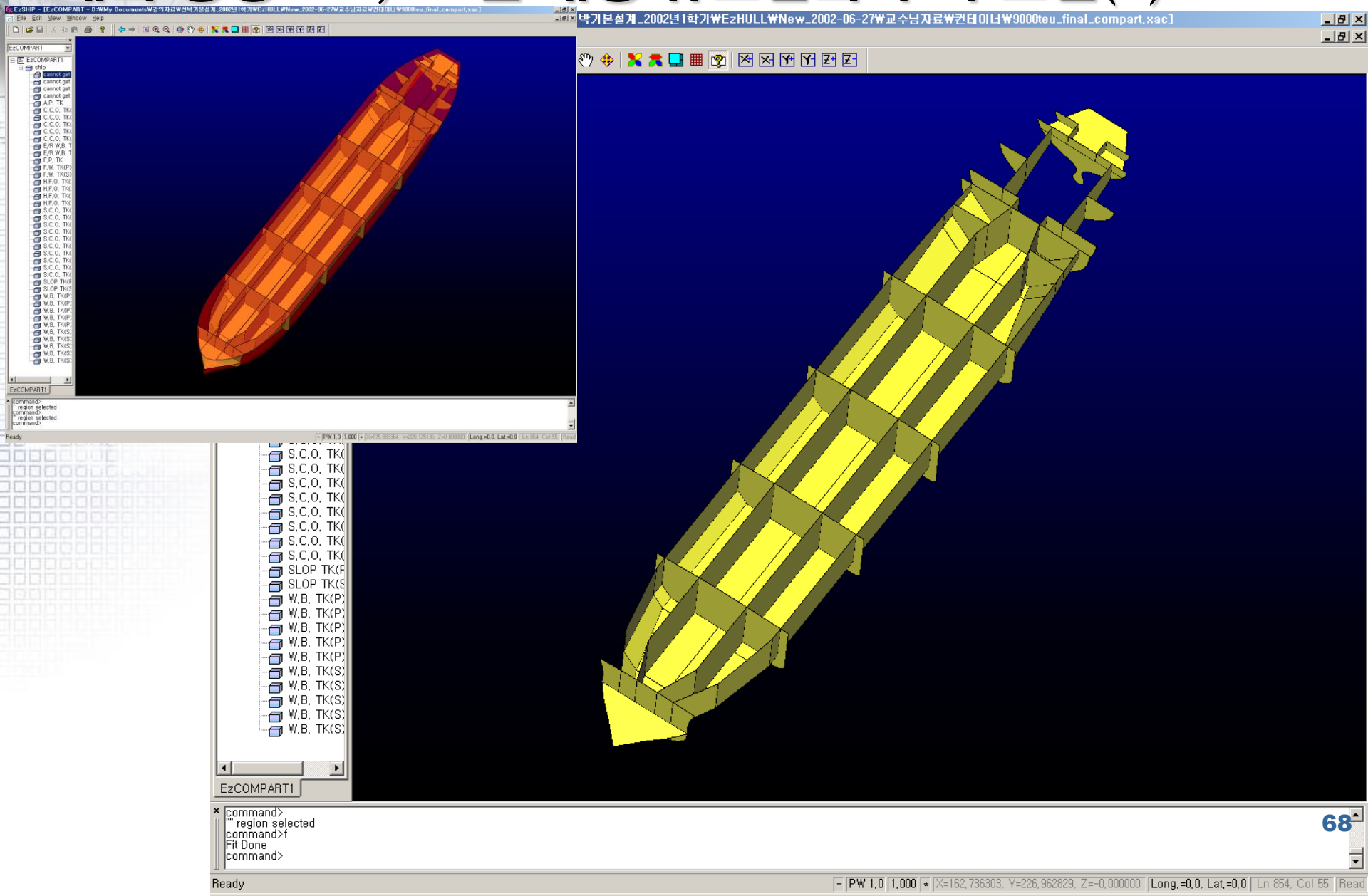




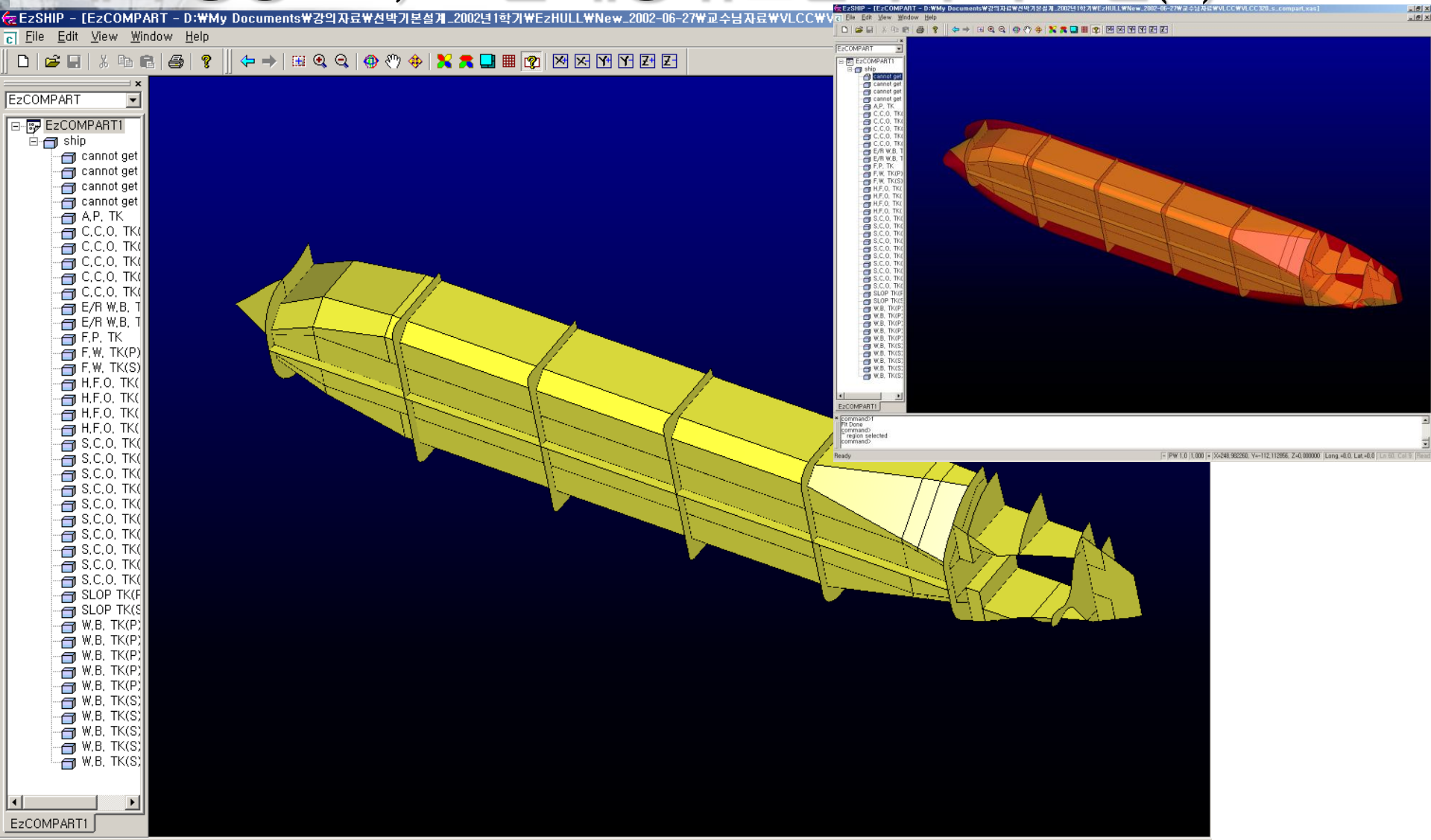
4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 구획 모델(3)



4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 구획 모델(4)

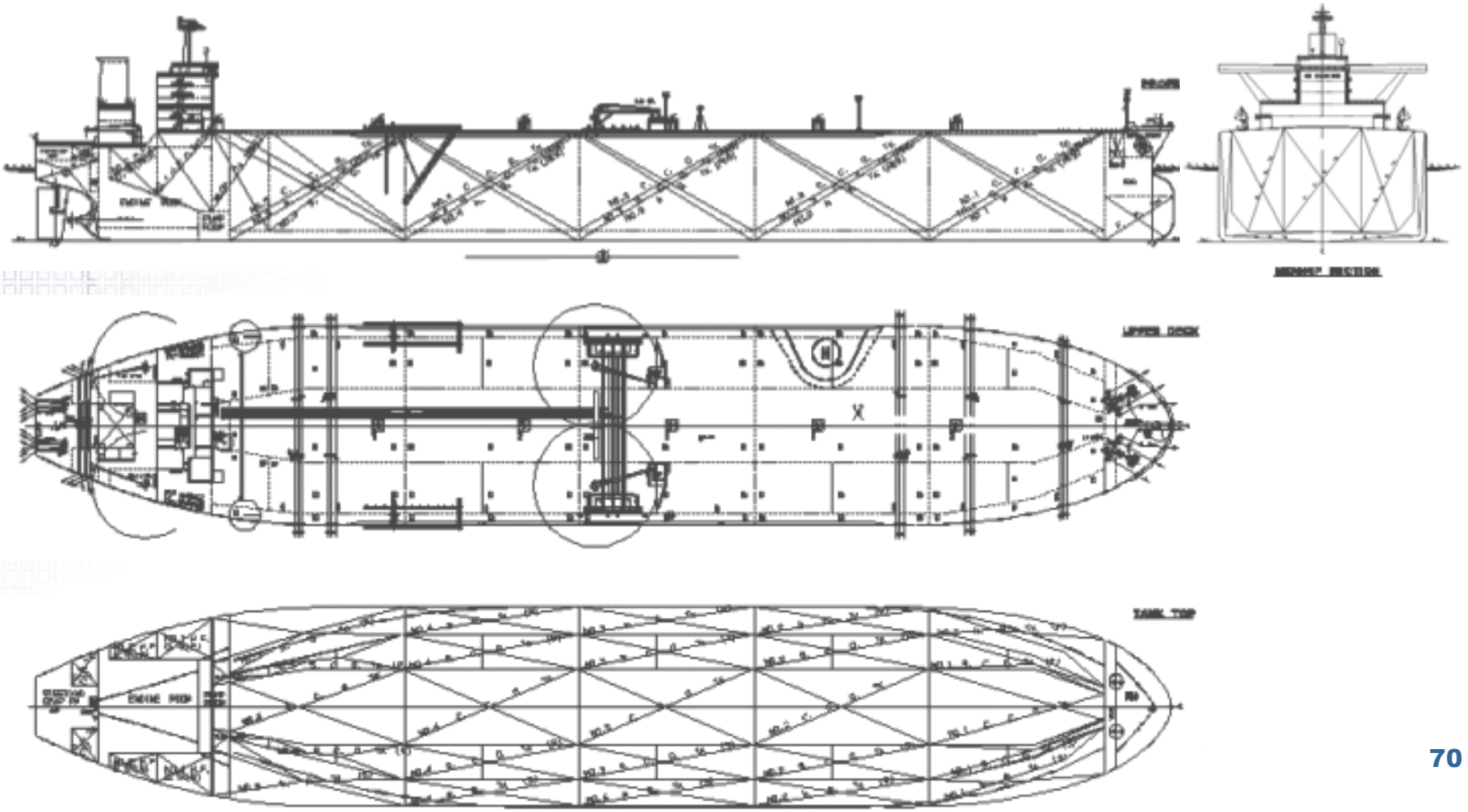


4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 구획 모델(5)



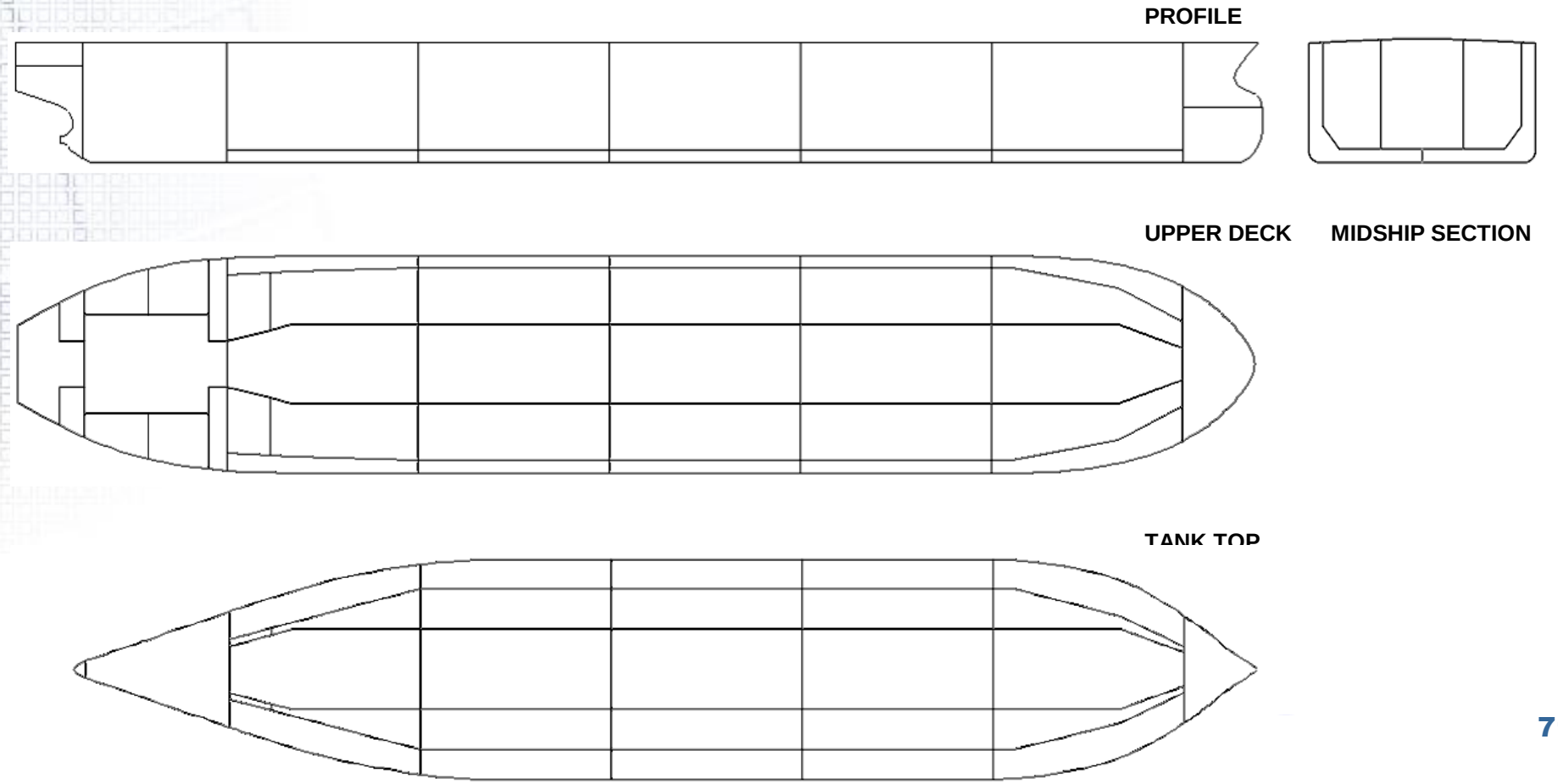
4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 G/A 생성(1)

재화 중량 320,000톤 대형 유조선의 일반 배치도



4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 G/A 생성(2)

3차원 선박 구획 모델로부터 자동 생성된 개략 일반 배치도(Sketch G/A)



4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 선박 계산(1)

선박 유체 정역학적 계산 예

Result Summary

DRAFT M	DISPL M	DISPL SW	DISPL FW	LCF	LCB	VCB	TPC	MTC/...	KMT	IL/100	IT/100	WPA	W
1,0	13271	13603	13271	17,042	17,661	0,511	143,1	23,7	249,6	739014	33056	13958	14
1,5	20357	20866	20357	16,675	17,376	0,768	147,2	25,1	172,5	782853	34963	14363	14
2,0	27618	28308	27618	16,427	17,159	1,027	150,3	26,2	132,9	817373	36410	14667	15
2,5	35012	35887	35012	16,189	16,979	1,285	152,7	27,1	108,2	845999	37434	14896	15
3,0	42507	43570	42507	15,961	16,819	1,544	154,5	27,9	91,4	870258	38211	15078	15
3,5	50085	51337	50085	15,772	16,675	1,802	156,1	28,5	79,4	890924	38874	15230	16
4,0	57734	59177	57734	15,595	16,543	2,060	157,5	29,1	70,4	908963	39463	15363	16
4,5	65445	67081	65445	15,459	16,423	2,318	158,7	29,6	63,4	924760	40000	15479	17
5,0	73211	75041	73211	15,301	16,313	2,576	159,7	30,1	57,9	938997	40480	15582	17
5,5	81025	83051	81025	15,172	16,209	2,834	160,6	30,5	53,3	951242	40915	15673	17
6,0	88881	91103	88881	15,072	16,112	3,092	161,5	30,8	49,6	961771	41311	15752	18
6,5	96774	99194	96774	14,989	16,024	3,349	162,1	31,1	46,4	970462	41666	15819	18
7,0	104701	107318	104701	14,854	15,941	3,607	162,8	31,4	43,7	979374	41998	15885	18
7,5	112659	115476	112659	14,682	15,858	3,864	163,5	31,7	41,4	988206	42309	15948	19
8,0	120648	123664	120648	14,478	15,774	4,121	164,1	31,9	39,4	996928	42590	16008	19
8,5	128667	131883	128667	14,246	15,686	4,379	164,7	32,2	37,7	1005690	42858	16066	19
9,0	136714	140132	136714	13,972	15,593	4,636	165,3	32,5	36,2	1014658	43114	16124	20
9,5	144790	148410	144790	13,654	15,494	4,893	165,8	32,8	34,8	1023573	43347	16179	20
10,0	152893	156716	152893	13,300	15,388	5,151	166,4	33,1	33,6	1032913	43572	16236	20
10,5	161026	165051	161026	12,898	15,272	5,408	167,0	33,4	32,6	1042768	43794	16293	21
11,0	169187	173417	169187	12,456	15,147	5,666	167,6	33,7	31,7	1053124	44016	16353	21
11,5	177379	181813	177379	11,988	15,012	5,924	168,2	34,1	30,9	1063920	44231	16414	22
12,0	185601	190241	185601	11,489	14,867	6,182	168,9	34,4	30,1	1075086	44440	16475	22
12,5	193854	198700	193854	10,939	14,712	6,440	169,5	34,8	29,5	1086935	44652	16539	22
13,0	202140	207193	202140	10,362	14,545	6,699	170,2	35,2	28,9	1099326	44862	16604	23
13,5	210459	215720	210459	9,747	14,368	6,958	170,9	35,6	28,4	1112421	45070	16672	23
14,0	218811	224282	218811	9,085	14,179	7,217	171,6	36,1	27,9	1125912	45277	16740	23
14,5	227199	232879	227199	8,394	13,978	7,477	172,3	36,5	27,5	1140266	45483	16812	24
15,0	235624	241514	235624	7,673	13,766	7,737	173,1	37,0	27,1	1155148	45688	16885	24
15,5	244085	250187	244085	6,899	13,541	7,997	173,8	37,5	26,8	1170743	45894	16960	24
16,0	252584	258898	252584	6,095	13,304	8,258	174,6	38,0	26,5	1186707	46101	17036	25
16,5	261121	267649	261121	5,274	13,055	8,520	175,4	38,5	26,3	1202800	46309	17112	25
17,0	269695	276437	269695	4,460	12,795	8,781	176,1	39,0	26,0	1218202	46518	17185	26
17,5	278305	285263	278305	3,698	12,525	9,043	176,9	39,5	25,8	1233139	46727	17255	26
18,0	286950	294123	286950	3,023	12,249	9,306	177,6	40,0	25,7	1247289	46934	17322	26
18,5	295627	303018	295627	2,415	11,969	9,568	178,2	40,4	25,5	1260786	47138	17387	27
19,0	304336	311944	304336	1,892	11,688	9,831	178,8	40,8	25,4	1272992	47338	17446	27
19,5	313073	320900	313073	1,438	11,408	10,094	179,4	41,1	25,3	1284654	47534	17503	28
20,0	321838	329884	321838	1,025	11,131	10,357	179,9	41,5	25,2	1295442	47722	17556	28

Hydrostatic Calculation Result Summary

D:₩My Documents₩강의자료₩선박기본설계_2002년1학기₩EzHU...

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

뒤로 검색 즐겨찾기 미디어

주소(D) D:₩My Documents₩강의자료₩선박기본설계_2002년 이동 연결

XML File Output

```
- <Hydro>
  - <HYDRO_TABLE no="1">
      <DRAFT_MOULDED>1.0</DRAFT_MOULDED>
      <DRAFT_EXTREAME>1.0</DRAFT_EXTREAME>
      <TRIM>0.0</TRIM>
      <HEEL>0.0</HEEL>
      <VOLUME_MOULDED>13271</VOLUME_MOULDED>
      <DISPL_TOT_SW>13603</DISPL_TOT_SW>
      <DISPL_TOT_FW>13271</DISPL_TOT_FW>
      <DISPL_AFT_BODY>5681</DISPL_AFT_BODY>
      <DISPL_FORE_BODY>7922</DISPL_FORE_BODY>
      <LCF_FROM_MIDSHIP>17.042</LCF_FROM_MIDSHIP>
      <LCB_FROM_MIDSHIP>17.661</LCB_FROM_MIDSHIP>
      <LCB_AFTBODY>-52.900</LCB_AFTBODY>
      <LCB_FOREBODY>68.266</LCB_FOREBODY>
      <VCB_ABOVE_BL>0.511</VCB_ABOVE_BL>
      <TPC>143.1</TPC>
      <MTC>2367.2</MTC>
      <KMT>249.6</KMT>
      <I_LONG>73901396</I_LONG>
      <I_TRAN>3305589</I_TRAN>
      <WPA_MOULDED>13958</WPA_MOULDED>
      <WETTED_SURFACE>14104</WETTED_SURFACE>
      <CB>0.6912</CB>
      <CP>0.7193</CP>
```

완료 내 컴퓨터

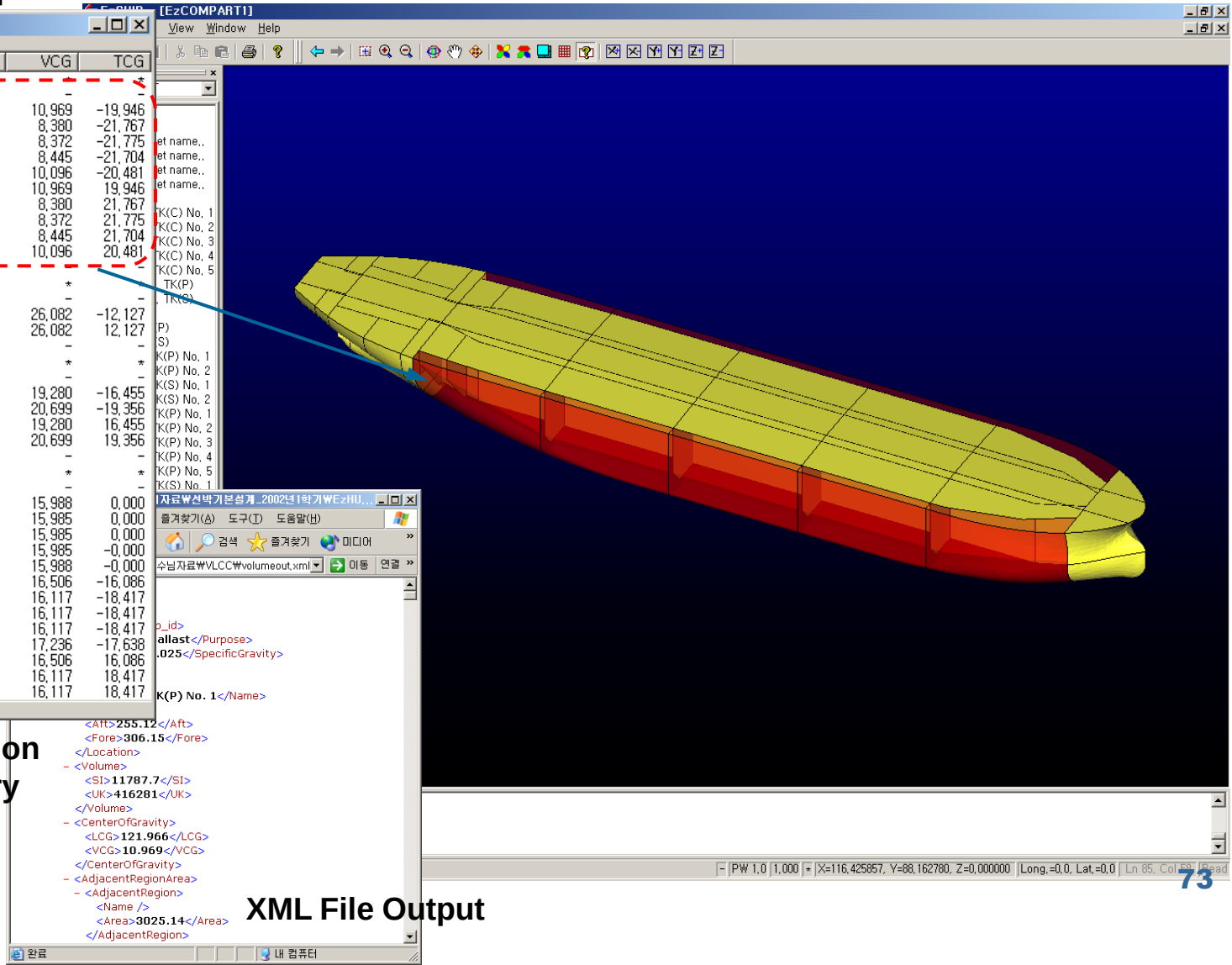
4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 선박 계산(2)

구획 용적 계산 예

Result Summary

COMPARTMENT	VOLUME	LCG	VCG	TCG
*	*	*	*	*
Water Ballast	1,025	-	-	-
W.B. TK(P) No. 1	11787.7	121.966	10.969	-19.946
W.B. TK(P) No. 2	9629.3	69.582	8.380	-21.767
W.B. TK(P) No. 3	9640.0	18.575	8.372	-21.775
W.B. TK(P) No. 4	9539.3	-32.276	8.445	-21.704
W.B. TK(P) No. 5	10511.0	-84.313	10.096	-20.481
W.B. TK(S) No. 1	11787.7	121.966	10.969	19.946
W.B. TK(S) No. 2	9629.3	69.582	8.380	21.767
W.B. TK(S) No. 3	9640.0	18.575	8.372	21.775
W.B. TK(S) No. 4	9539.3	-32.276	8.445	21.704
W.B. TK(S) No. 5	10511.0	-84.313	10.096	20.481
TOTAL	102214.9	-	-	-
*	*	*	*	*
Fresh Water	1,000	-	-	-
F.W. TK(P)	427.5	-150.487	26.082	-12.127
F.W. TK(S)	427.5	-150.487	26.082	12.127
TOTAL	855.1	-	-	-
*	*	*	*	*
Heavy Fuel Oil	0.980	-	-	-
H.F.O. TK(P) No. 1	2002.5	-111.526	19.280	-16.455
H.F.O. TK(P) No. 2	3057.8	-121.472	20.699	-19.356
H.F.O. TK(S) No. 1	2002.5	-111.526	19.280	16.455
H.F.O. TK(S) No. 2	3057.8	-121.472	20.699	19.356
TOTAL	10120.6	-	-	-
*	*	*	*	*
Liquid Cargo 0.79	0.787	-	-	-
C.C.O. TK(C) No. 1	26005.0	118.502	15.988	0.000
C.C.O. TK(C) No. 2	28799.5	69.605	15.985	0.000
C.C.O. TK(C) No. 3	28799.5	18.575	15.985	0.000
C.C.O. TK(C) No. 4	28799.5	-32.455	15.985	-0.000
C.C.O. TK(C) No. 5	26805.0	-82.008	15.988	-0.000
S.C.O. TK(P) No. 1	14241.3	117.018	16.506	-16.086
S.C.O. TK(P) No. 2	19403.6	69.605	16.117	-18.417
S.C.O. TK(P) No. 3	19403.6	18.575	16.117	-18.417
S.C.O. TK(P) No. 4	19403.6	-32.455	16.117	-18.417
S.C.O. TK(P) No. 5	12899.6	-76.598	17.236	-17.638
S.C.O. TK(S) No. 1	14241.3	117.018	16.506	16.086
S.C.O. TK(S) No. 2	19403.6	69.605	16.117	18.417
S.C.O. TK(S) No. 3	19403.6	18.575	16.117	18.417

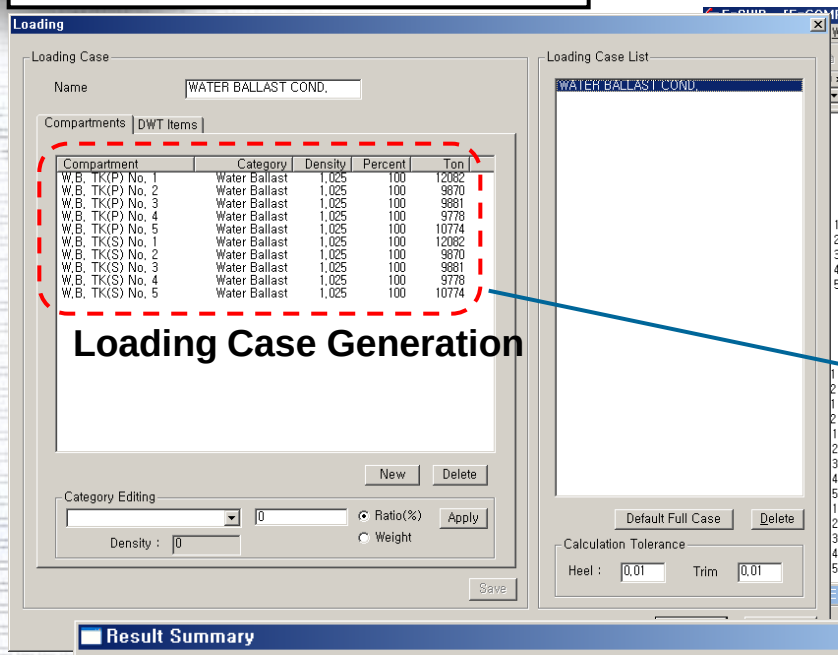
Volume Calculation Result Summary



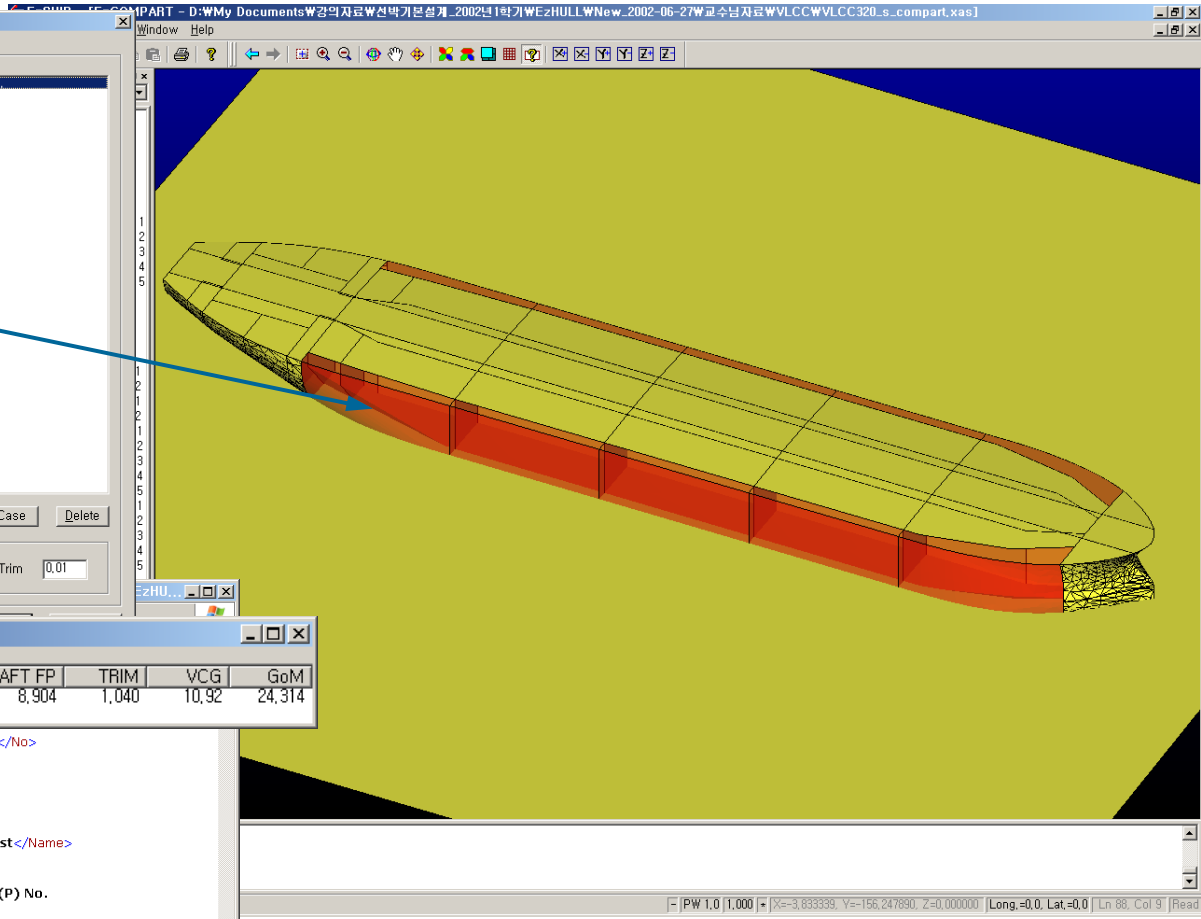
XML File Output

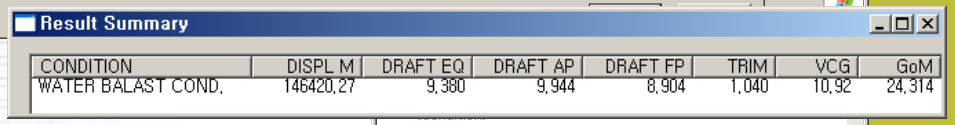
4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 선박 계산(3)

비손상 시 복원성 계산 예

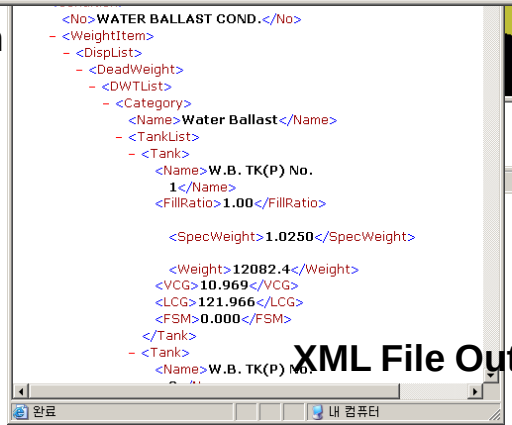


Loading Case Generation





Loading Calculation Result Summary



XML File Output

4. 구획 모델 완성
- 재화 중량 320,000톤 대형 유조선의 선박 계산(4)

손상 시 복원성 계산 예

Result Summary

CONDITION	DISPL M	DRAFT EQ	TRIM	HEEL	LCG	KG	GM
Damage case scant	123581.29	7.995	3.342	-9.873	6.789	10.138	36.351

Damage Calculation Result Summary

D:₩My Documents₩강의자료₩선박기본설계_2002년1학기₩EzHU...

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

주소(D) ...2002-06-27₩교수님자료₩VLCC₩damagegout.xml

```
- <DamageCondition>
  - <Condition>
      <No>Damage case scant</No>
    - <HydroResult>
        <Equiv_Draft>7.995</Equiv_Draft>
        <Trim>3.342</Trim>
        <HeelAng>-9.873</HeelAng>
        <Displacement>123581.29</Displacement>
        <LCG>6.789</LCG>
        <VCG>10.138</VCG>
        <GM>36.351</GM>
      </HydroResult>
    - <Stability>
      - <Case>
          <HeelAngle>0.000</HeelAngle>
          <GZValue>3.702</GZValue>
          <HeelDisp>120567.107</HeelDisp>
          <HeelDraft>7.995</HeelDraft>
          <HeelTrim>0.000</HeelTrim>
        </Case>
      - <Case>
          <HeelAngle>0.100</HeelAngle>
          <GZValue>3.743</GZValue>
          <HeelDisp>120567.190</HeelDisp>
          <HeelDraft>7.995</HeelDraft>
```

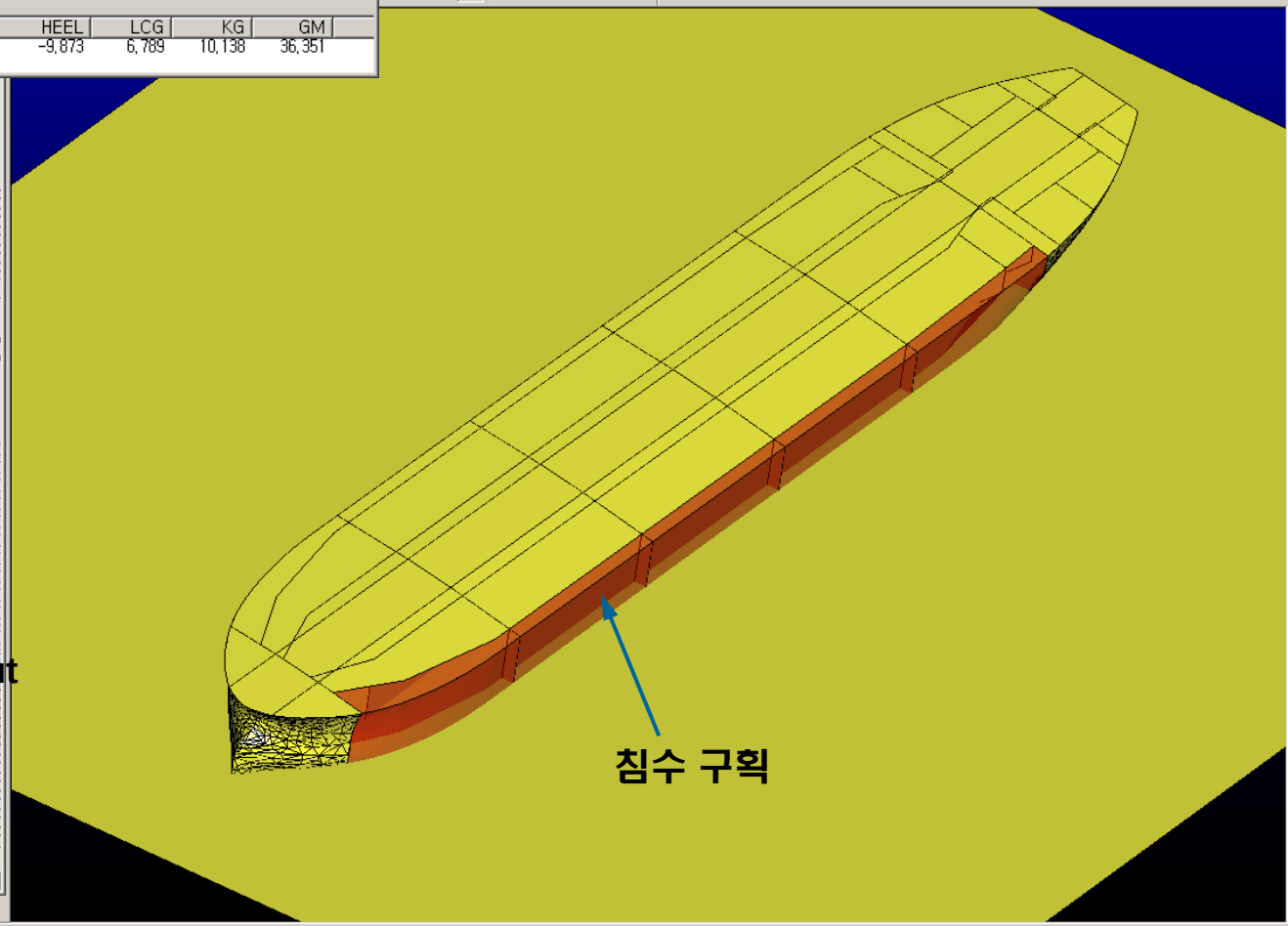
XML File Output

EzSHIP - [EzCOMPART - D:₩My Documents₩강의자료₩선박기본설계_2002년1학기₩EzHULL₩New_2002-06-27₩교수님자료₩VLCC₩VLCC320_s_compart.xas]

File Edit View Window Help

ship
cannot get
cannot get
cannot get
cannot get
A.P. TK
W.B. TK(P)
W.B. TK(S)
cannot get nar

EzCOMPART1



command>
"W.B. TK(P) No. 4" region selected
command>
"W.B. TK(P) No. 5" region selected
command>

Ready

4. 구획 모델 완성
- 화물창 용적 145,000CBM LNG선의 구획 모델(1)

Principal Dimensions

항목		값	비고
주요제원	LOA	282.6 m	
	LBP	271.6 m	
	B	43.4 m	
	D	26.5 m	
	Td / Ts	11.3 / 12.0 m	
Cargo Capacity		145,216 CBM	at Td
Speed		20.2 knots	at Td
Main Engine	Type	Mitsubishi MS 40-2	
	MCR	38,709 PS x 83.0 rpm	
	NCR	34,838 PS x 80.0 rpm	
Propeller Diameter		8.28 m	