Resource Allocation Rule Modeling Case Study

Use Case

Assign ships to appropriate berths at a port based on matching ship size and requirements to berth size and capabilities.

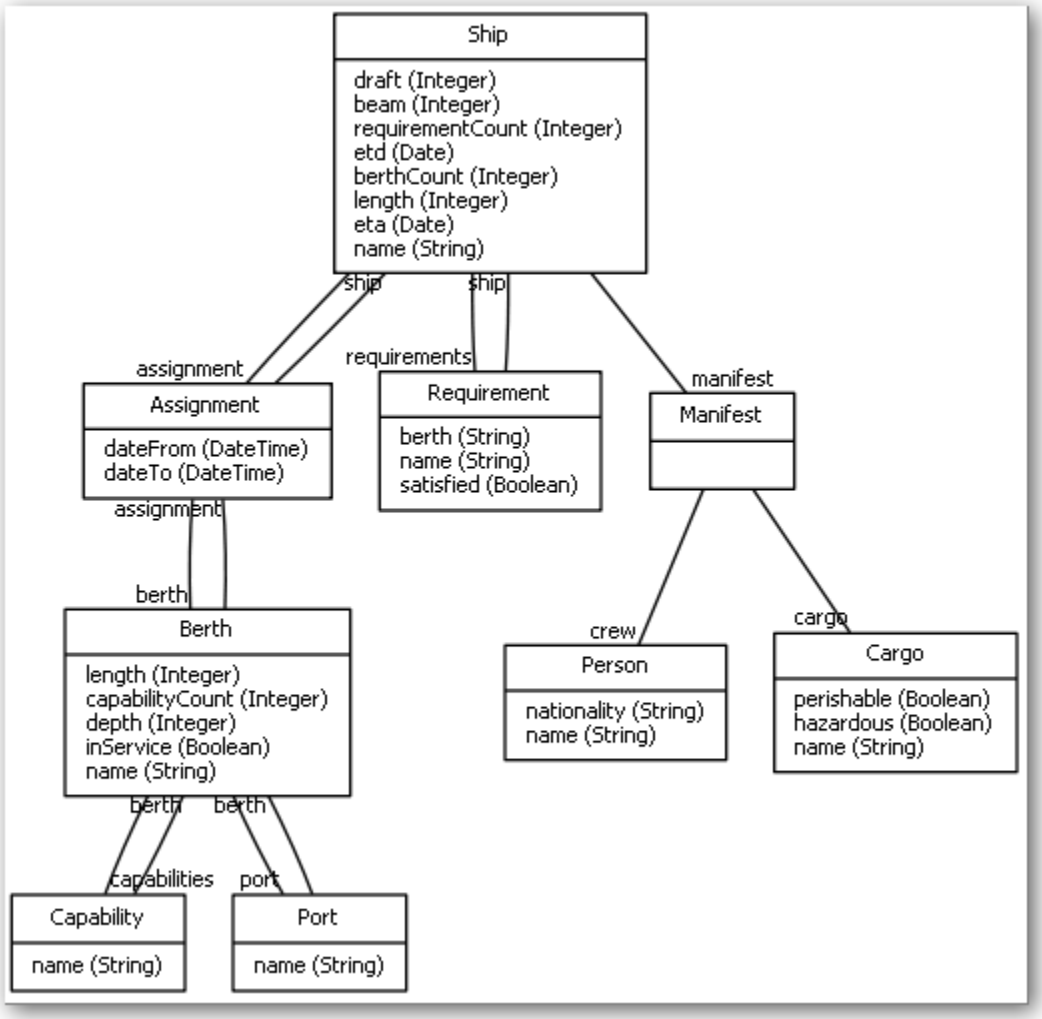
The Rules

A ship must fit in the berth (both length and draft)
The berth must satisfy all the requirements of the ship.
The berth must be in service.
Ships that have only one possible berth assignment take priority over ships that have other choices.
Ships with perishable cargo can bump those without (when they are both competing for the same berth).
If two ships require the same berth and both have perishable cargo then human intervention is required.
Ships with perishable cargo need to be cleared by customs. (see rule model “Mangoes from Pakistan”)

Ignore the dates when the ships require the berth for this example.
For extra points take into account the dates when berths are required.

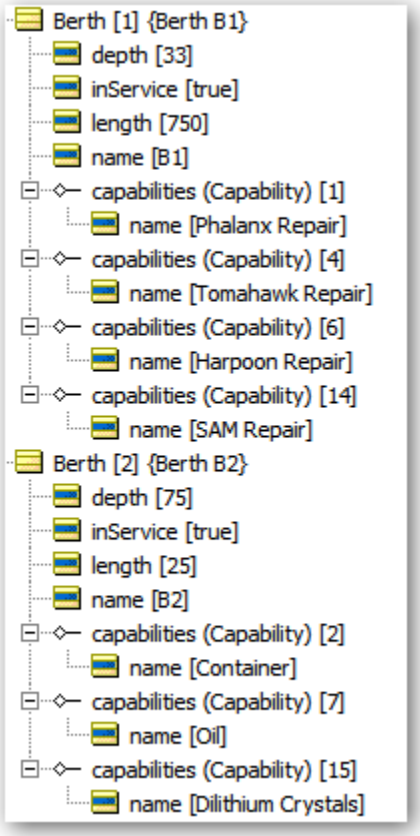
Data Model

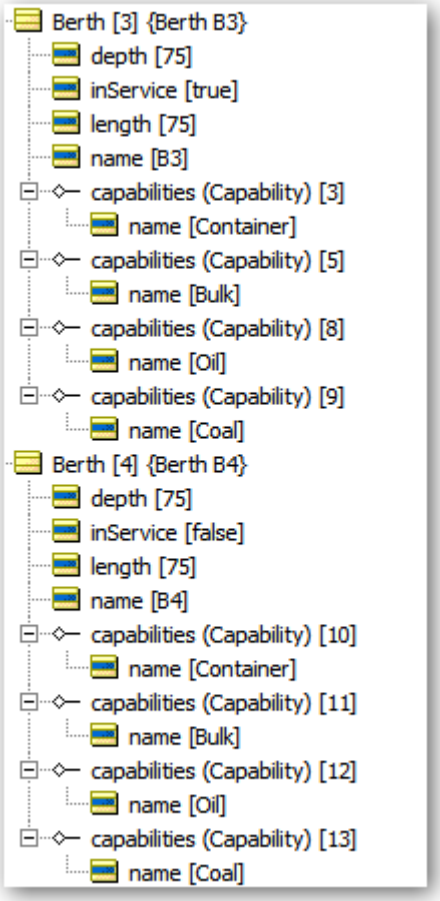
Just a suggestion. Feel free to create your own data model.

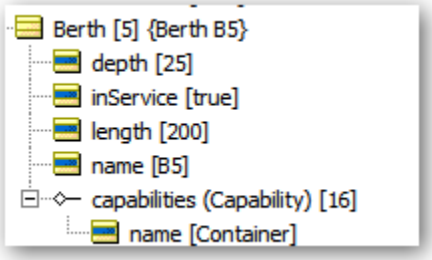


Sample Input Data

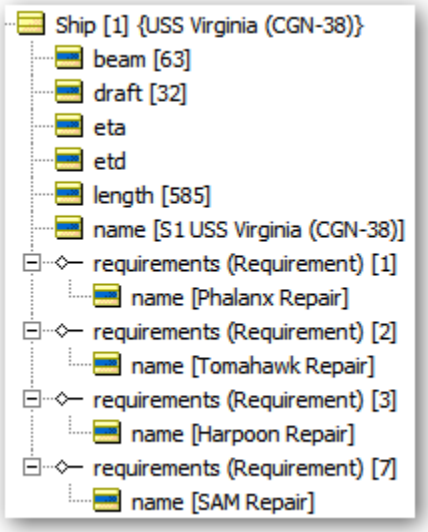
Berths

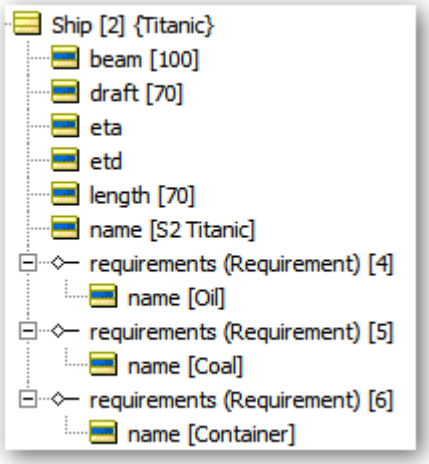


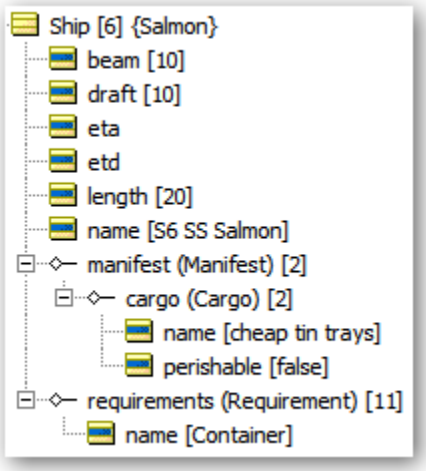


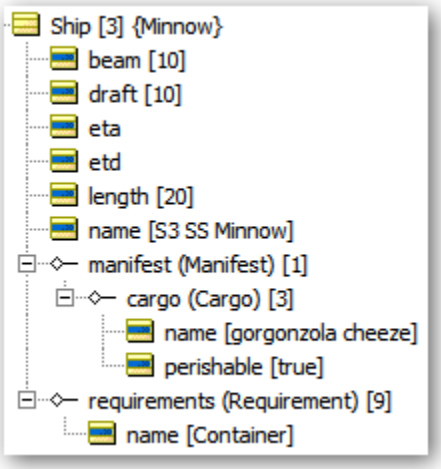


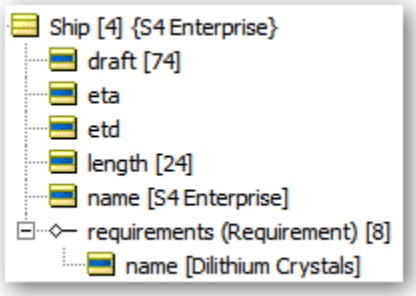
Ships

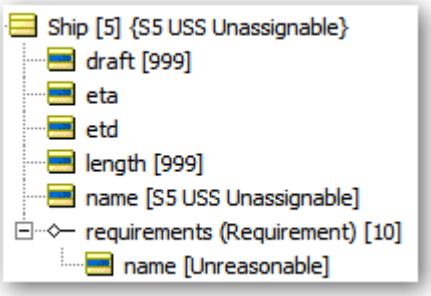

Ship [1] {USS Virginia (CGN-38)}
beam [63]
draft [32]
eta
etd
length [585]
name [S1 USS Virginia (CGN-38)]
requirements (Requirement) [1]
name [Phalanx Repair]
requirements (Requirement) [2]
name [Tomahawk Repair]
requirements (Requirement) [3]
name [Harpoon Repair]
requirements (Requirement) [7]
name [SAM Repair]


Ship [2] {Titanic}
beam [100]
draft [70]
eta
etd
length [70]
name [S2 Titanic]
requirements (Requirement) [4]
name [Oil]
requirements (Requirement) [5]
name [Coal]
requirements (Requirement) [6]
name [Container]


Ship [6] {Salmon}
beam [10]
draft [10]
eta
etd
length [20]
name [S6 SS Salmon]
manifest (Manifest) [2]
cargo (Cargo) [2]
name [cheap tin trays]
perishable [false]
requirements (Requirement) [11]
name [Container]


Ship [3] {Minnow}
beam [10]
draft [10]
eta
etd
length [20]
name [S3 SS Minnow]
manifest (Manifest) [1]
cargo (Cargo) [3]
name [gorgonzola cheeze]
perishable [true]
requirements (Requirement) [9]
name [Container]


Ship [4] {S4 Enterprise}
draft [74]
eta
etd
length [24]
name [S4 Enterprise]
requirements (Requirement) [8]
name [Dilithium Crystals]

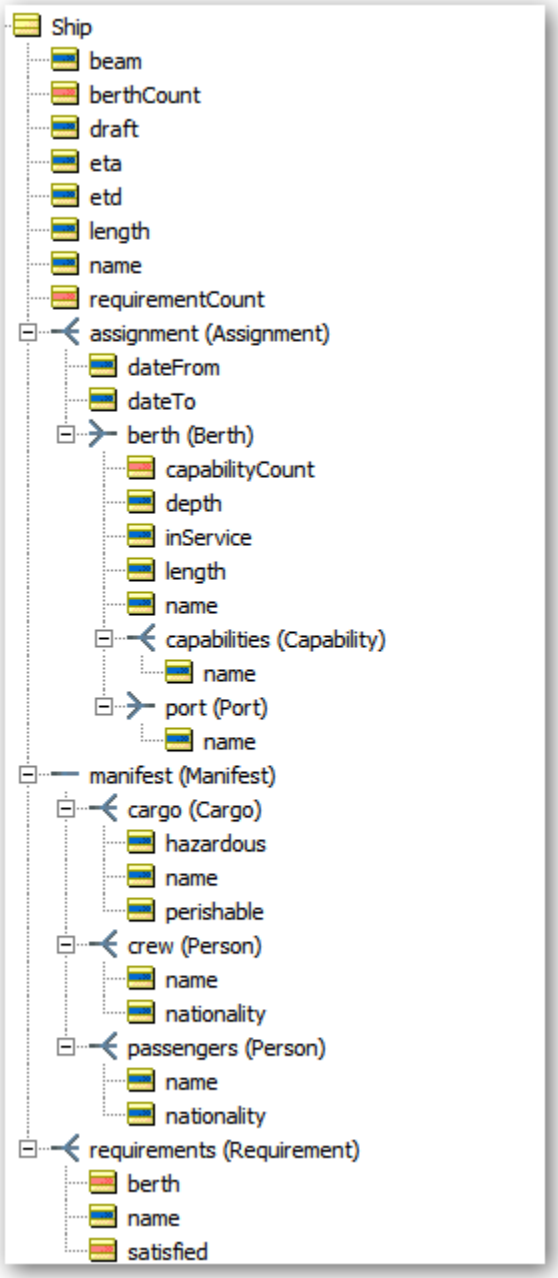

Ship [5] {S5 USS Unassignable}
draft [999]
eta
etd
length [999]
name [S5 USS Unassignable]
requirements (Requirement) [10]
name [Unreasonable]

Expected Results

Severity	Message	Entity
Violation	B1: Berth does not have all the capabilities required by S6 SS Salmon	Berth[1]
Violation	B1: Berth does not have all the capabilities required by S3 SS Minnow	Berth[1]
Violation	B3: Berth does not have all the capabilities required by S4 Enterprise	Berth[3]
Info	S1 USS Virginia (CGN-38): Berth B1 has all the capabilities required by the ship	Ship[1]
Info	S1 USS Virginia (CGN-38) has only one possible berth B1	Ship[1]
Info	Final: S1 USS Virginia (CGN-38) has only one possible berth B1	Ship[1]
Info	S2 Titanic: Berth B3 has all the capabilities required by the ship	Ship[2]
Info	S2 Titanic has only one possible berth B3	Ship[2]
Info	Final: S2 Titanic has only one possible berth B3	Ship[2]
Info	S3 SS Minnow: Berth B2 has all the capabilities required by the ship	Ship[3]
Info	S3 SS Minnow: Berth B3 has all the capabilities required by the ship	Ship[3]
Info	S3 SS Minnow: Berth B5 has all the capabilities required by the ship	Ship[3]
Info	S3 SS Minnow can be assigned to any of 3 berths	Ship[3]
Info	We can remove B2 from S3 SS Minnow since it has other possible berths and B2 is the only berth possible for S4 Enterprise	Ship[3]
Info	We can remove B3 from S3 SS Minnow since it has other possible berths and B3 is the only berth possible for S2 Titanic	Ship[3]
Info	Final: S3 SS Minnow has only one possible berth B5	Ship[3]
Warning	Perishable goods needs to be inspected by the FDA before S3 SS Minnow can dock	Ship[3]
Info	S4 Enterprise: Berth B2 has all the capabilities required by the ship	Ship[4]
Info	S4 Enterprise has only one possible berth B2	Ship[4]
Info	Final: S4 Enterprise has only one possible berth B2	Ship[4]
Warning	S5 USS Unassignable has no assigned berth	Ship[5]
Warning	Final: S5 USS Unassignable has no assigned berth	Ship[5]
Info	S6 SS Salmon: Berth B2 has all the capabilities required by the ship	Ship[6]
Info	S6 SS Salmon: Berth B3 has all the capabilities required by the ship	Ship[6]
Info	S6 SS Salmon: Berth B5 has all the capabilities required by the ship	Ship[6]
Info	S6 SS Salmon can be assigned to any of 3 berths	Ship[6]
Info	We can remove B3 from S6 SS Salmon since it has other possible berths and B3 is the only berth possible for S2 Titanic	Ship[6]
Info	We can remove B2 from S6 SS Salmon since it has other possible berths and B2 is the only berth possible for S4 Enterprise	Ship[6]
Info	Ships with perishable cargo can bump those without (when they are both competing for the same berth) S3 SS Minnow bumps S6 SS Salmon	Ship[6]
	Final: S6 SS Salmon has no assigned berth	Ship[6]
	S6 SS Salmon is cleared for docking	Ship[6]

NOTE: if a ship has no berth then it should not be cleared for docking

Vocabulary


Ship
beam
berthCount
draft
eta
etd
length
name
requirementCount
assignment (Assignment)
dateFrom
dateTo
berth (Berth)
capabilityCount
depth
inService
length
name
capabilities (Capability)
name
port (Port)
name
manifest (Manifest)
cargo (Cargo)
hazardous
name
perishable
crew (Person)
name
nationality
passengers (Person)
name
nationality
requirements (Requirement)
berth
name
satisfied

The Basic Allocation Decision Table

Natural Language View

Conditions	0	1	2	3
is this particular ship requirement satisfied by one of the berth's capabilities?		T	-	-
are all the ships requirements satisfied		-	T	F
Actions				
Post Message(s)	✉		✉	✉
initially mark each of the ship's requirements as NOT satisfied	☑			
count the number of capabilities that each berth provides	☑			
count the number of requirements for each ship	☑			
flag this requirement as satisfied		T		
create a new assignment of the ship to the berth			☑	

Implementation View

Conditions	0	1	2	3
requirement.name = capability.name		T	-	-
requirement -> forAll(satisfied)		-	T	F
Actions				
Post Message(s)	✉		✉	✉
requirement.satisfied = false	☑			
aBerth.capabilityCount = capability->size	☑			
aShip.requirementCount = requirement->size	☑			
requirement.satisfied		T		
Assignment.new[berth=aBerth,ship=aShip]			☑	

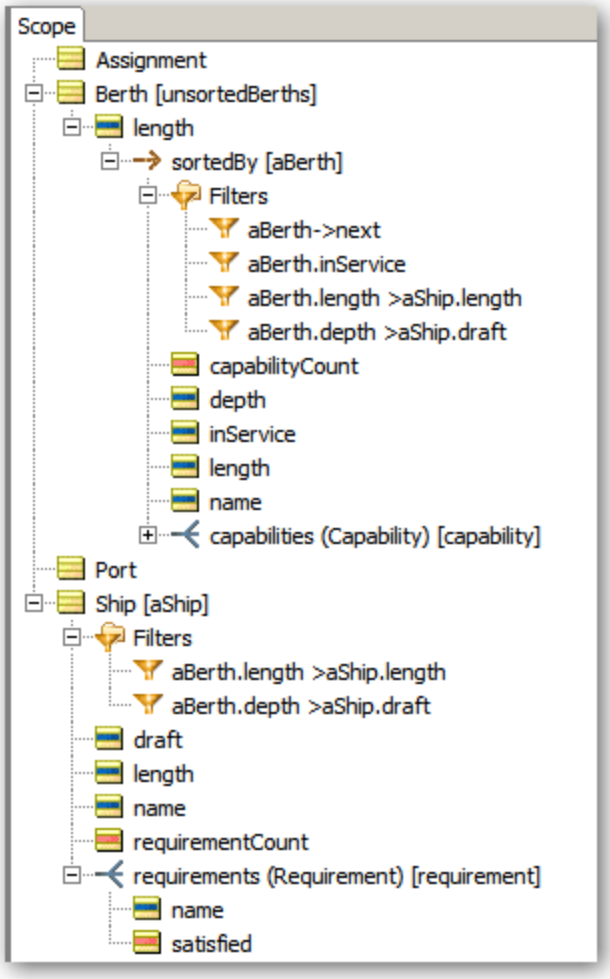
Note: Rule 1 will check on all requirements before rule 2 checks if they were all satisfied.

Filters

These ensure that only berths that are in service and are big enough for the ship are considered.

Filters	
1	aBerth->next
2	aBerth.length >aShip.length
3	aBerth.depth >aShip.draft
4	aBerth.inService

Line 1 simply takes the berths in ascending order of size to make sure we don't allocate a tiny ship to a very large berth thus depriving the bigger ship of a berth.



Berths are sorted by length

Ancillary Rule Sheet for Resolving Conflicts

When two ships are assigned to the same berth

Conditions	0	1	2	3
How many possible berths are there for ship 1?		1	1	1
How many possible berths are there for ship 2?		1	> 1	1
Do ship 1 and ship 2 share an assigned berth?		T	T	T
Does ship 1 have a perishable cargo?		T	-	T
Does ship2 have a perishable cargo?		F	-	T
Actions				
Post Message(s)	✉	✉	✉	✉
Count the number of berths that are assigned to ship 1	☑			
Count the number of berths that are assigned to ship 2	☑			
Remove the berth from ship 2's assignment.		☑	☑	

Rule Statements

Ref	Text
A0	Ships that have only one possible berth assignment displace ships that have other choices
1	Ships with perishable cargo can bump those without (when they are both competing for the same berth) {ship1.name} bumps {ship2.name}
2	We can remove {ship2.assignment.berth.name} from {ship2.name} since it has other possible berths and {ship2.assignment.berth.name} is the only berth possible for {ship1.name}
3	Conflict - both ships require same berth and both have perishable cargo. Human intervention required.

Implementation

Scope

- Ship [ship1]
- Ship [ship2]

Filters

1	ship1<>ship2
2	
3	
4	
5	
6	

	Conditions	0	1	2	3
a	ship1.berthCount		1	1	1
b	ship2.berthCount		1	> 1	1
c	ship1.assignment.berth = ship2.assignment.berth		T	T	T
d	s1Cargo->exists(perishable)		T	-	T
e	s2Cargo->exists(perishable)		F	-	T
f					
	Actions				
	Post Message(s)	✉	✉	✉	✉
A	ship1.berthCount = s1berths->size	☑			
B	ship2.berthCount = s2berths->size	☑			
C	ship2.assignment.remove		☑	☑	

Ancillary rule sheets for Customs

Some possible customs rules (courtesy of John Masefield)

	Conditions	1	2	3	4	5	6	7	8
a	Description of goods	'Cheap Tin Trays'	-	other	'Tyne coal'	'Road-rail'	'pig-lead'	'Firewood'	'iron-ware'
b	Are the goods perishable?	-	T	F	-	-	-	-	-
	Actions								
	Post Message(s)	✉	✉	✉	✉	✉	✉	✉	✉
A									
	Overrides								

Rule Statements | Rule Messages | Natural Language | Properties

Ref	Text
1	Cheap Tin Trays are subject to an import tax of $1 per tray - must be manually counted before {Ship.name} can dock
2	Perishable goods needs to be inspected by the FDA before {Ship.name} can dock
3	{Ship.name} is cleared for docking
4	Tyne coal is subject to EPA pollution tax
5	Road rail imports need special clearance
6	Pig lead imports are limited to 1 million tonnes per year
7	Firewood may only be imported if it comes from non-endangered tree species
8	Iron ware is subject to import tax

Alternative natural language view of the Assignment rules

Scope

- Berth [unsortedBerths]
  - length
- Ship [aShip]
  - name
  - requirements (Requirement) [requirement]

Filters

1	laat beschikbare aanlegplaatsen in aanmerking komen startend met de kleinste.
2	aanlegplaasten komen alleen in aanmerking als ze groter zijn dan het schip
3	laat alleen aanlegplaatsen in aanmerking komen als de diepte groter is dan de diepgang van het schip
4	alleen aanlegplaatsen die operationeel zijn in aanmerking laten komen.

	Conditions	0	1	2	3
a	komt een van de mogelijkheden van deze aanlegplaat tegemoed aan de vereisten van het schip?		'yes'		
b	worden alle vereisten van het schip afgedekt?	-	-	'yes'	'no'
c					
	Actions				
	Post Message(s)				
A	markeer alle vereisten waaran het schip dient te voldoen als NIET VOLDAAN	☑	☐	☐	☐
B	tel het aantal mogenlijkheden dat elke aanlegplaats biedt	☑	☐	☐	☐
C	tel het aantal vereisten van elk schip	☑	☐	☐	☐
D	markeer deze vereiste als VOLDAAN	☐	☑	☐	☐
E	wijs dit schip toe aan de aanlegplaats	☐	☐	☑	☐
F					
G					
H					
I					