Container Terminals

An Initial Domain Analysis & Description Sketch

Dines Bjørner
Fredsvej 11, DK-2840 Holte and DTU, DK-2800 Kgs. Lyngby, Denmark.
e-mail: bjorner@gmail.com, URL: www.imm.dtu.dk/~dibj

. **November 6, 2018: 01:50 am**

The ECNU November 2018 Course Project

Abstract

- We present a recording of stages and steps
- of a development of a domain analysis & description
- of an answer to he question:
- *what, mathematically, is a container terminal?*

√ Course Project I √

Course Project II

Course Project III

Abstract

- This is a report on an experiment.
  - At any stage of development,
  - and the present draft stage is judged 2/3 "completed"
  - it reflects how I view an answer to the question
  - *what is a container terminal port ?*
  - mathematically speaking.

1. Introduction

TO BE WRITTEN

1.1. Survey of Literature on Container-related Matters

- [1, A Container Line Industry Domain, 2007]
- [2, A-Z Dictionary of Export, Trade and Shipping Terms]
- [3, Portworker Development Programme: PDP Units]
- [4, An interactive simulation model for the logistics planning of container operations in seaports,1996]
- [5, Stowage planning for container ships to reduce the number of shifts, 1998]
- [6, Container stowage planning: a methodology for generating computerised solutions, 2000]
- [7, Container ship stowage problem: complexity and connection to the coloring of circle graphs, 2000]

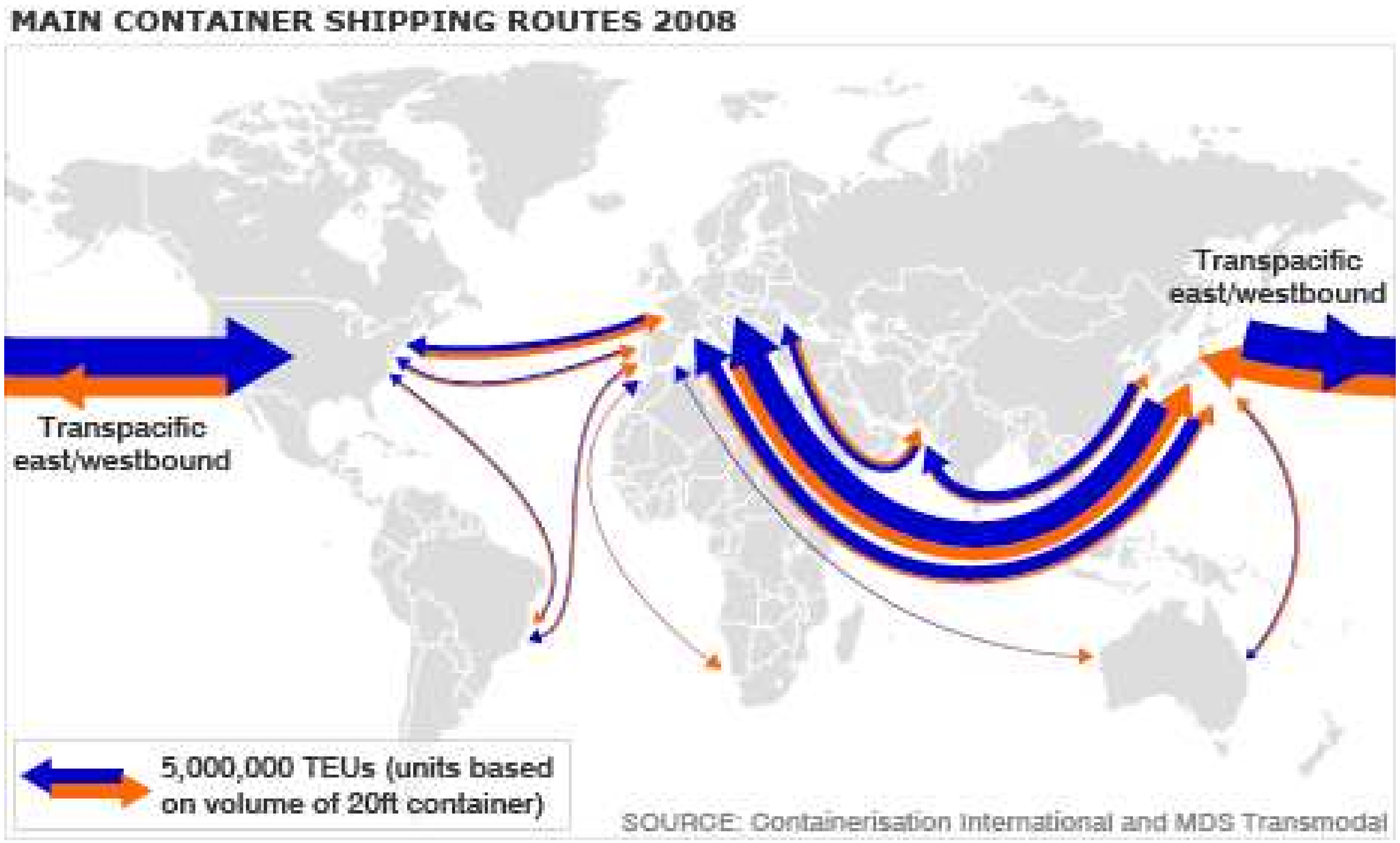
- [8, Container stowage pre-planning: using search to generate solutions, a case study, 2001]
- [9, A genetic algorithm with a compact solution encoding for the container ship stowage problem, 2002]
- [10, Multi-objective ... stowage and load planning for a container ship with container rehandle ..., 2004]
- [11, Container terminal operation and operations research - a classification and literature review, 2004]
- [12, Online rules for container stacking, 2010]

2. Some Pictures

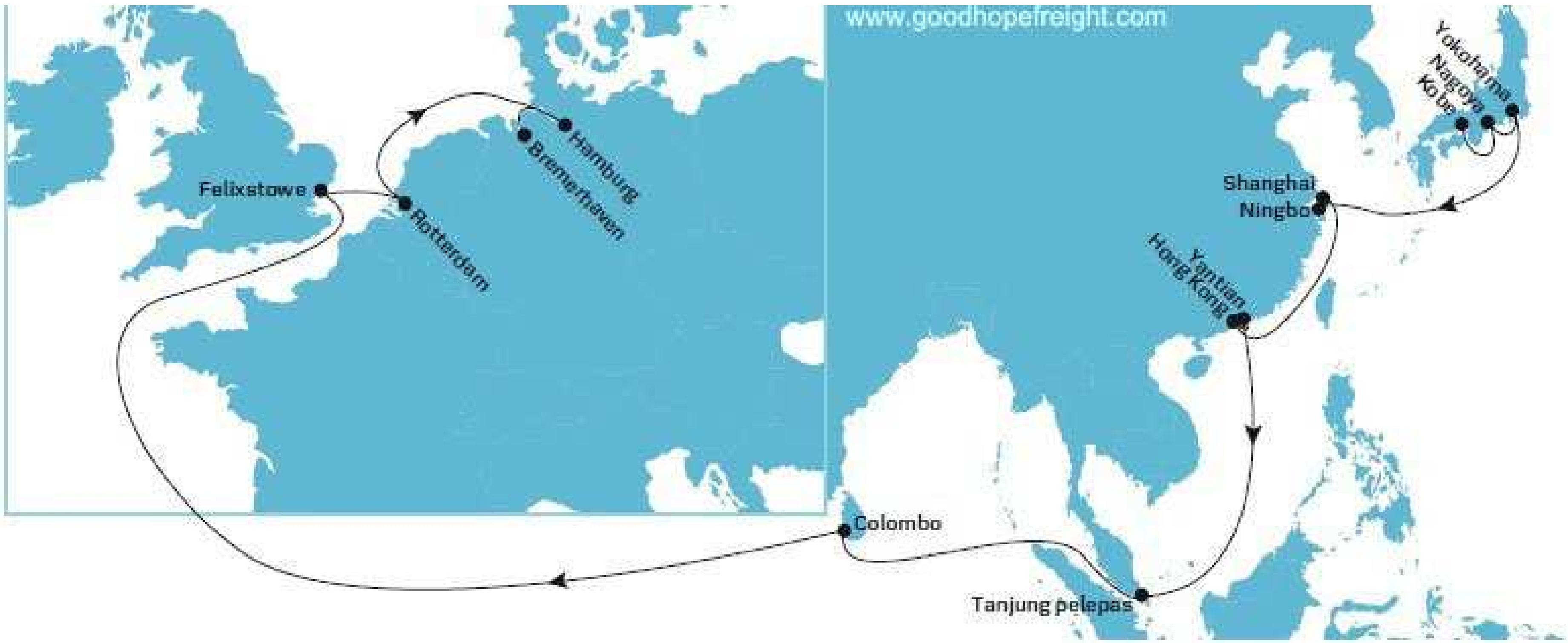
2.1. Terminal Port Container Stowage Area

2.2. Container Stowage Area and Quay Cranes

2.3. Container Vessel Routes



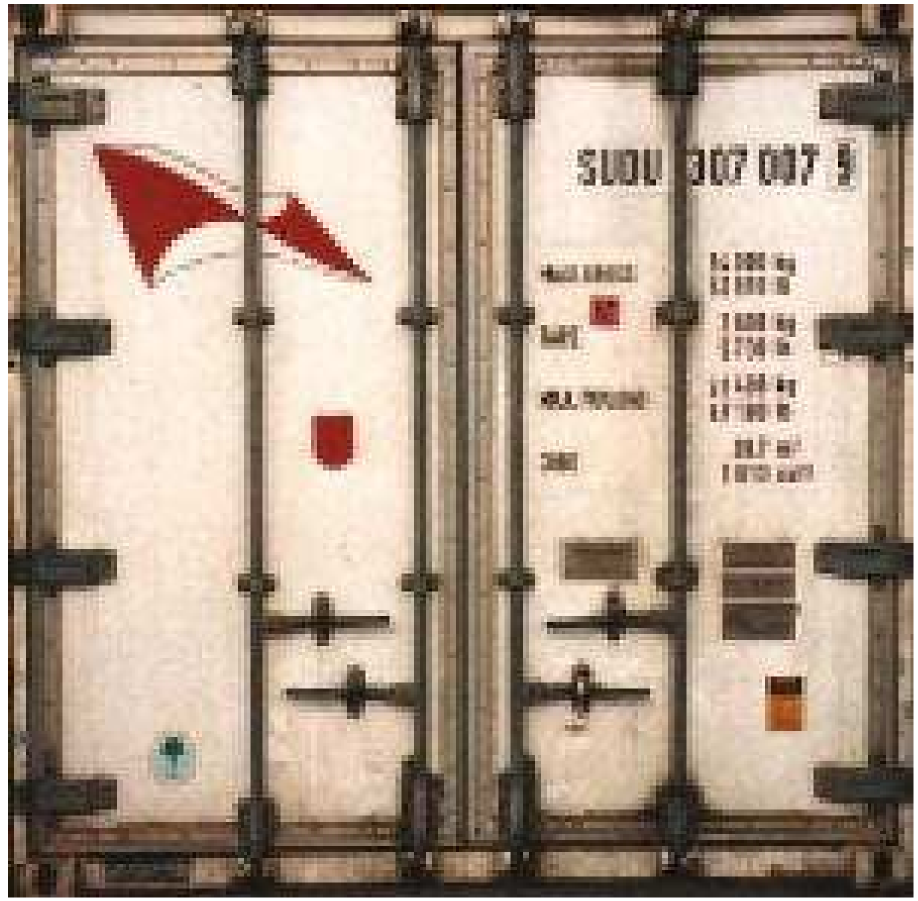

Southampton
Le Havre
Hamburg
Bremerhaven
Rotterdam
Zeebrugge
Malta
Tanger
Suez Canal
Jebel Ali
Khor Fakkan
Shanghai/Ningbo
Xiamen
Yantian/Chiwan
Hong Kong
Port Kelang


www.goodhopefreight.com
Felixstowe
Rotterdam
Bremerhaven
Hamburg
Yokohama
Nagoya
Kobe
Shanghai
Ningbo
Yantian
Hong Kong
Colombo
Tanjung pelepas

2.4. Containers

2.4.1. 40 and 20 Feet Containers

2.4.2. Container Markings

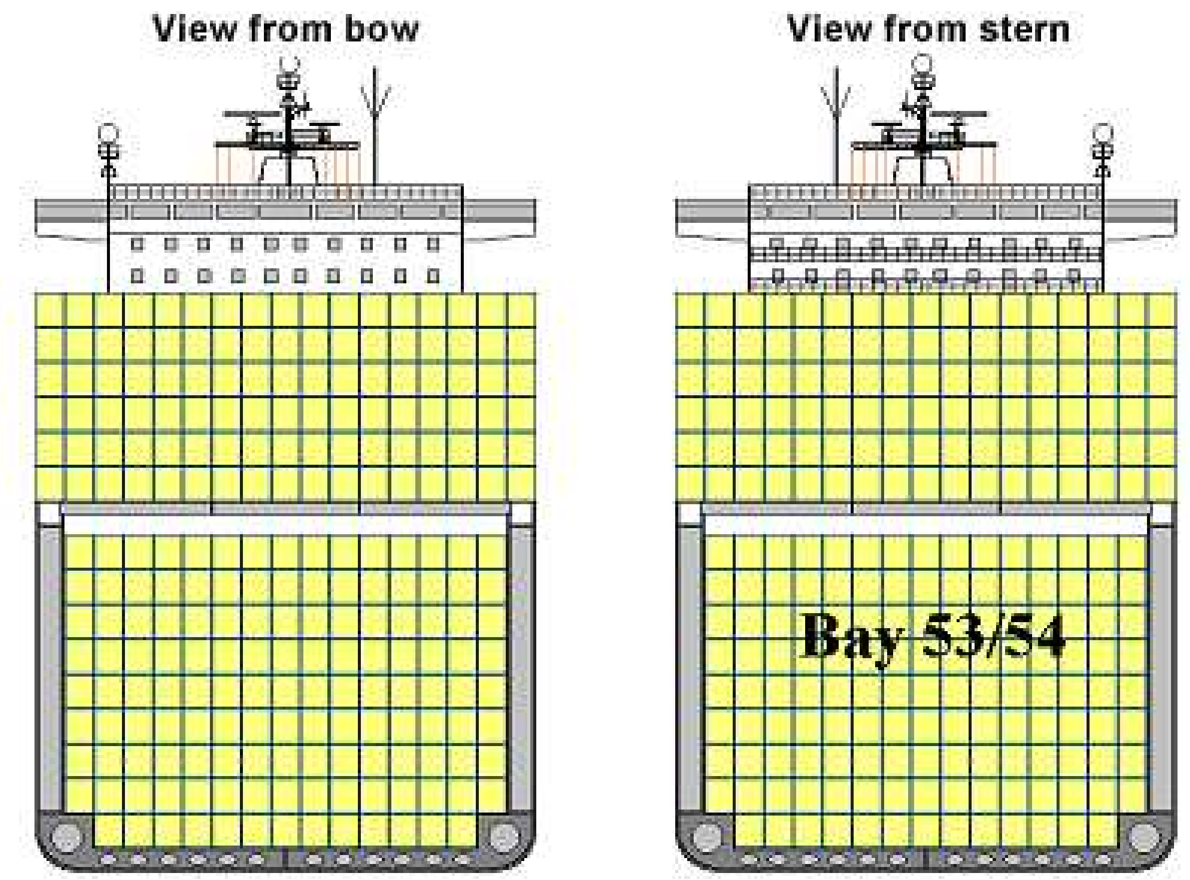


2.5. Container Vessels

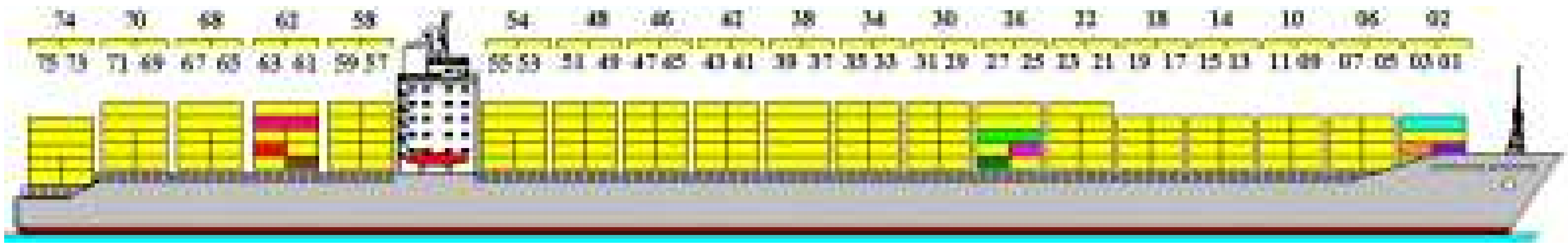
Quay cranes and *vessel* showing *row* of *aft (rear) bay*.

2.6. Container Stowage Area: Bays Rows, Stacks and Tier

Bay, Row, Tier Numbers. Row Numbers

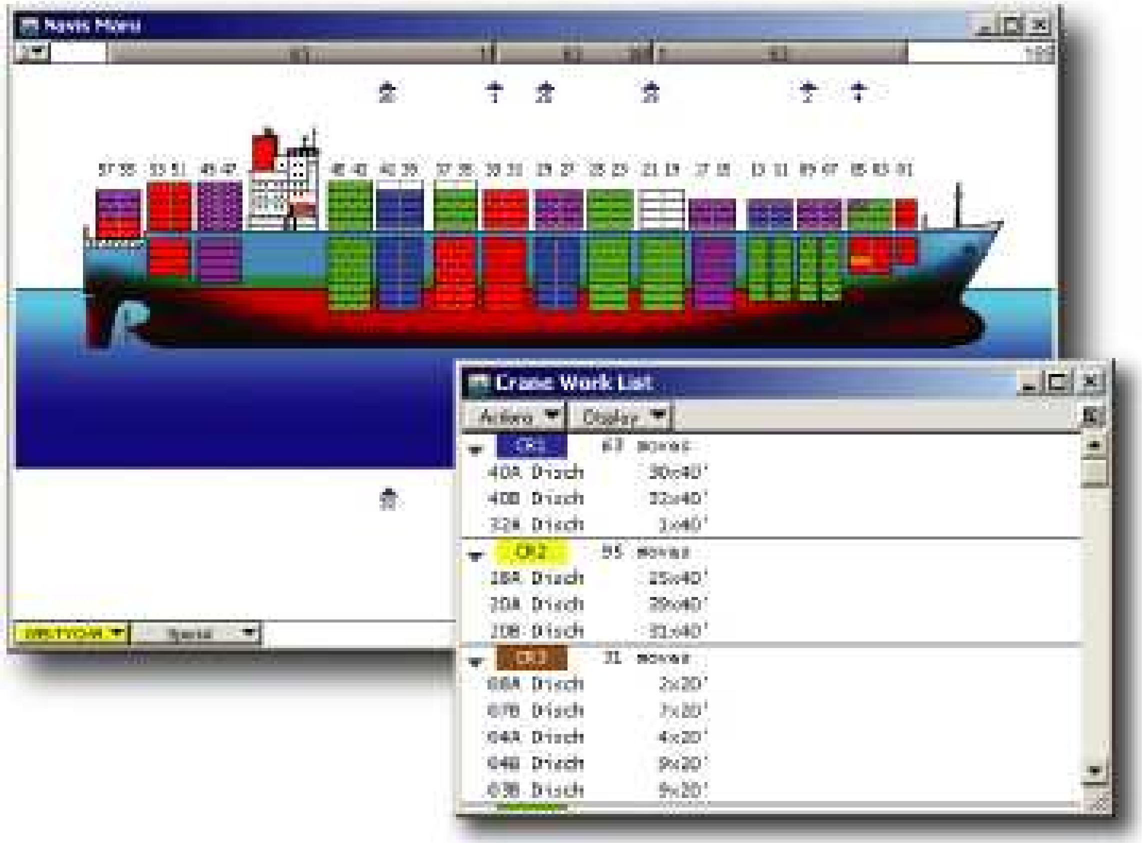

View from bow
View from stern
Bay 53/54

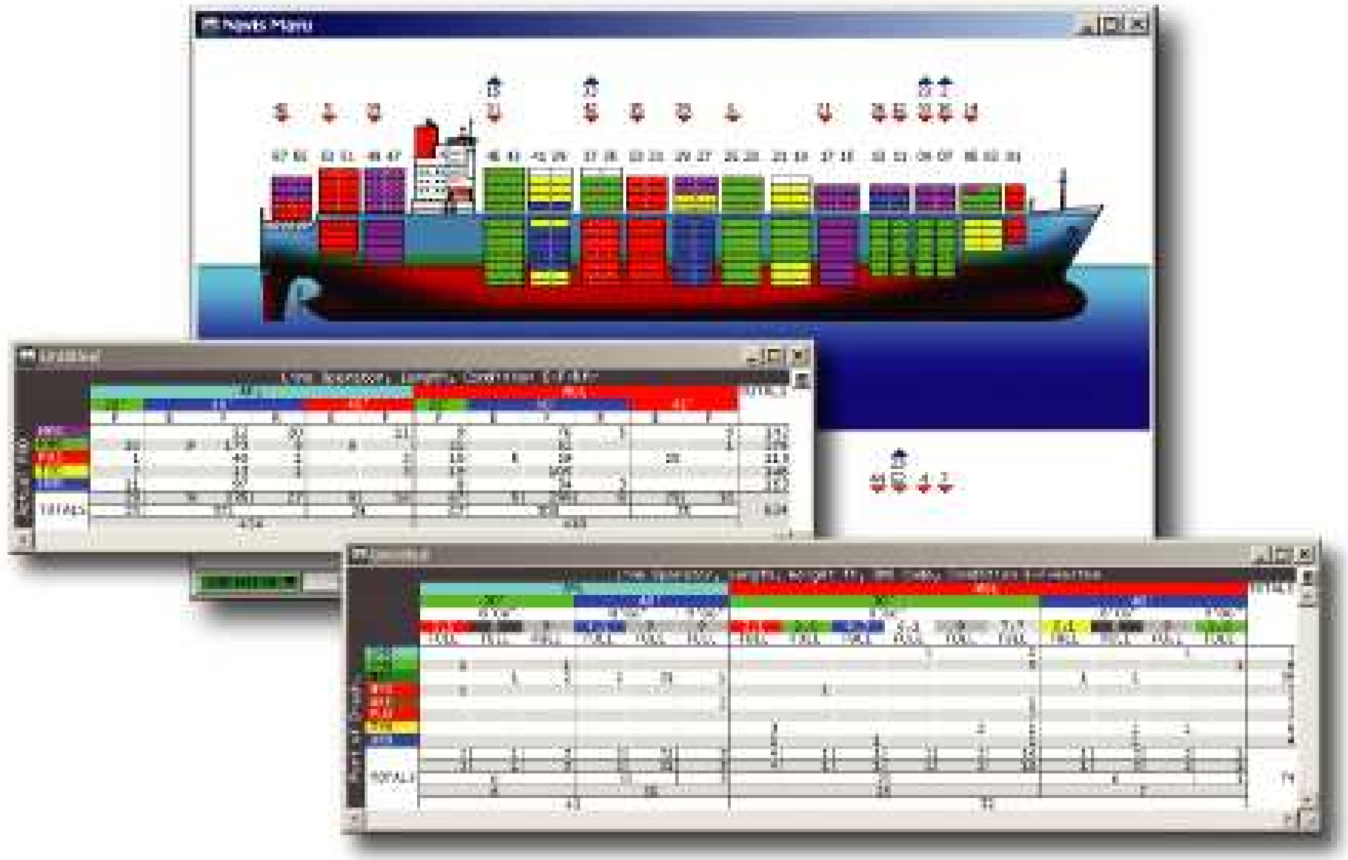
Cross section of a Bay. Tier Numbers.



Bay Numbering

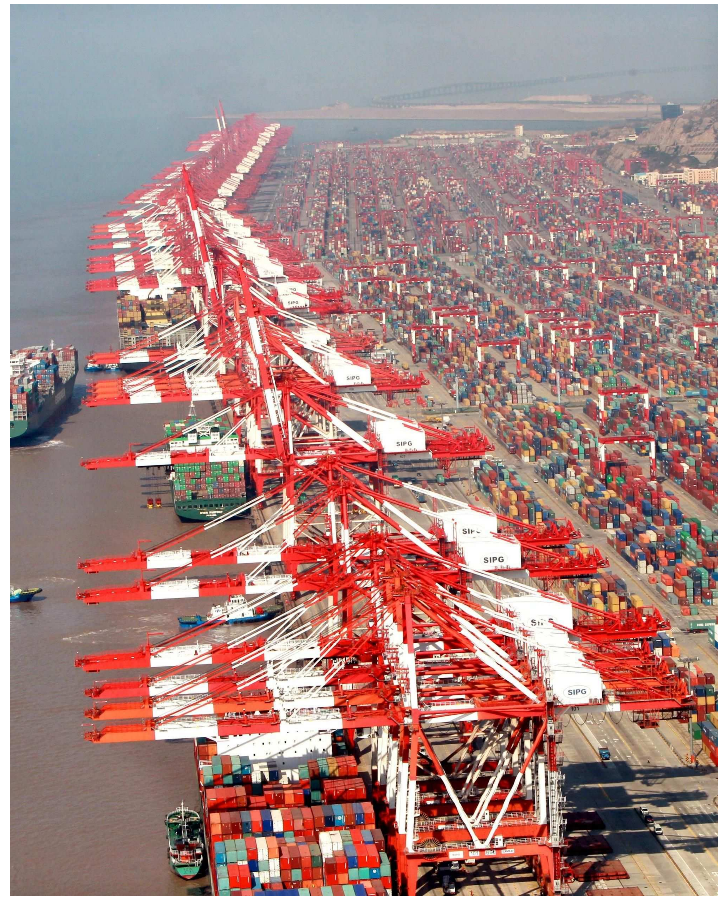
2.7. Stowage Software





2.8. Quay Cranes

2.9. Container Stowage Area and Stack Cranes




© Imaginechina/REX/Shutterstock

2.10. Container Stowage Area

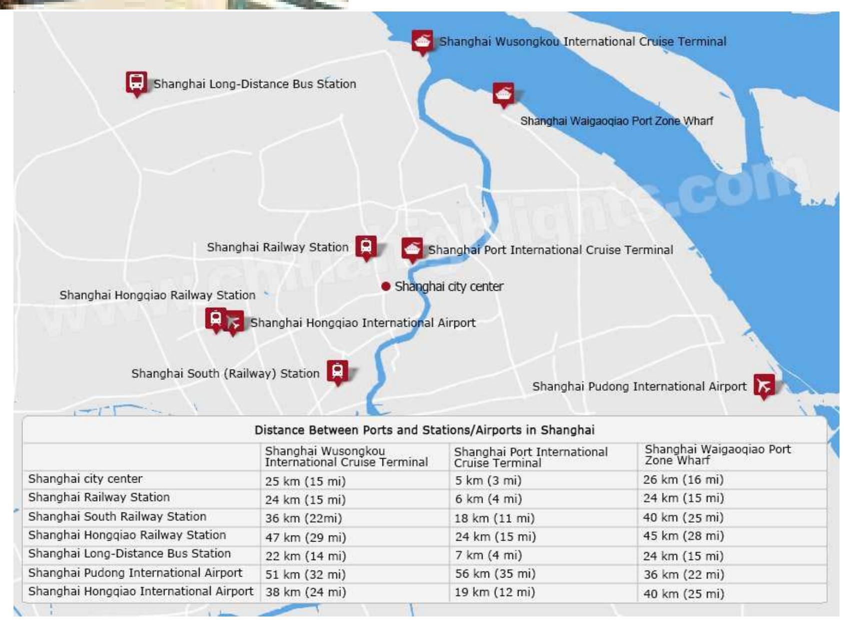
2.11. Quay Trucks

2.12. Map of Shanghai and YangShan

3. SECT

- *Shanghai East Container Terminal*
  - is the joint venture terminal of
  - *APM Terminals* and
  - *Shanghai International Port Group*
  - in *Wai Gao Qiao* port area of *Shanghai*.
- No.1 Gangjian Road, Pudong New District, Shanghai, China


CMA CGM



Distance Between Ports and Stations/Airports in Shanghai

	Shanghai Wusongkou International Cruise Terminal	Shanghai Port International Cruise Terminal	Shanghai Waigaoqiao Port Zone Wharf
Shanghai city center	25 km (15 mi)	5 km (3 mi)	26 km (16 mi)
Shanghai Railway Station	24 km (15 mi)	6 km (4 mi)	24 km (15 mi)
Shanghai South Railway Station	36 km (22mi)	18 km (11 mi)	40 km (25 mi)
Shanghai Hongqiao Railway Station	47 km (29 mi)	24 km (15 mi)	45 km (28 mi)
Shanghai Long-Distance Bus Station	22 km (14 mi)	7 km (4 mi)	24 km (15 mi)
Shanghai Pudong International Airport	51 km (32 mi)	56 km (35 mi)	36 km (22 mi)
Shanghai Hongqiao International Airport	38 km (24 mi)	19 km (12 mi)	40 km (25 mi)

4. Main Behaviours

- From consumer/origin to consumer/final destination:
  - **container loads** onto **land truck**;
  - **land truck travels** to **terminal stack**;
  - **container unloads** by means of **terminal stack crane** from **land truck** onto **terminal stack**.
  - **Container moves** from **stack** to **vessel**:
    - **terminal stack crane moves container** from **terminal stack** to **quay truck**,
    - **quay truck moves container** from **terminal stack** to **quay**,
    - **quay crane moves container** to **top of a vessel stack**;
  - **Container moves** on **vessel** from **terminal** to **terminal**:
    - Either container is unloaded at a next terminal port to a stack and from there to a container truck
    - or: container is unloaded at a next terminal port to a stack and from there to a next container vessel.

4.1. A Diagram

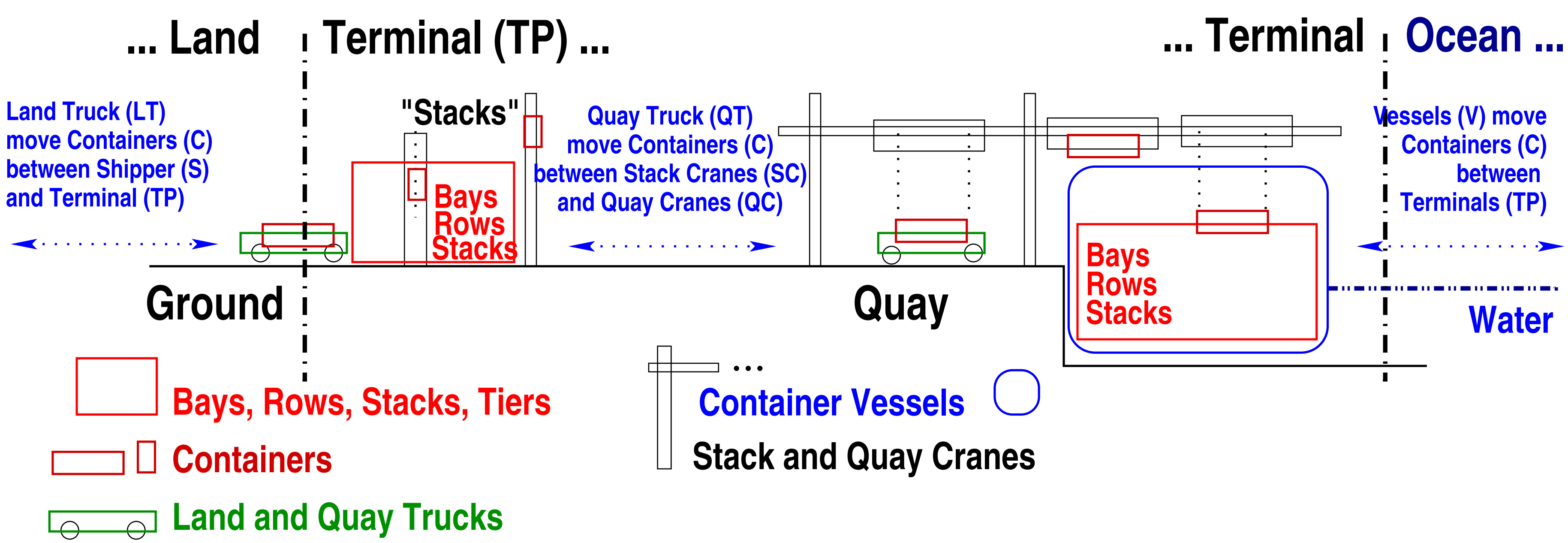


Fig. 1: Container Terminal Ports, I

A "from the side" snapshot of terminal port activities

4.2. Terminology - a Caveat

Bay [1]: contains indexed set of *rows* (of stacks of containers).

Container : smallest unit of central (i.e., huge) concern !

Container Stowage Area : An area of a vessel or a terminal where containers are stored, during voyage, respectively awaiting to be either brought out to shippers or onto vessels.

Crane :

- **Stack Crane** : moves *containers* between *land* or *terminal trucks* and *terminal stacks*.
- **Quay Crane** : moves *containers* between [*land* or] *terminal trucks* and *vessels*.

[1] The terms introduced in this section are mine. They are most likely not the correct technical terms of the container shipping and stowage trade. I expect to revise this section, etc.

Land : ... as you know it ...

Ocean : ... as you know it ...

Shipper : arranges shipment of containers with container lines

Quay : area of terminal next to vessels (hence water).

Row : contains indexed set of ***stacks*** (of containers).

Stack : contains indexed set of ***containers***.

We shall also, perhaps confusingly, use the term stack referring to the land-based bays of a terminal.

Terminal : area of land and water between land and ocean equipped with container stowage area, and stack and quay cranes, etc.

Truck :

Land Truck : privately operated truck
transport *containers* between *shippers* and *stack cranes*.

Quay Truck : terminal operated special truck
transport *containers* between *stack cranes* and *quay cranes*.

Tier : index of *container* in *stack*.

Vessel : contains a *container stowage area*.

4.3. Assumptions

- Without loss of generality we can assume that
  - there is exactly one stack crane per terminal stack;
  - quay cranes each serve exactly one bay on a vessel;
  - there are enough quay cranes to serve all bays of any berthed vessel;
  - quay trucks may serve any (quay and stack) crane;
  - land trucks may serve more than one terminal;
  - et cetera.

Course Project II

5. Endurants

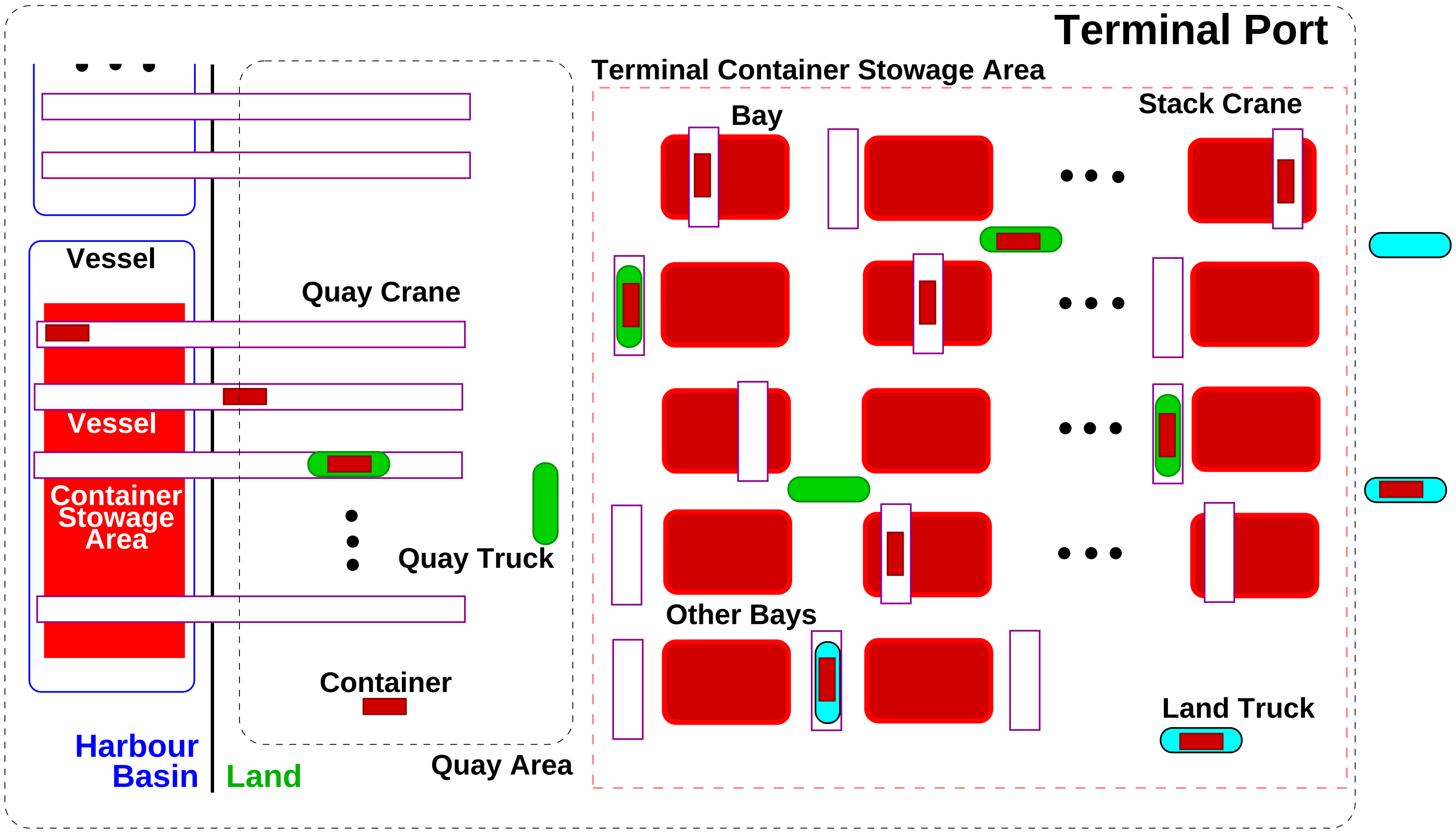


Fig. 2: Container Terminal Ports, II

A “from above” snapshot of terminal port activities

5.1. Parts

1 In the container line industry, CLI, we can observe

2 an indexed set
of two or more container ***terminal*** ports, TP;

3 an indexed set
of one or more container ***vessels***, V; and

4 a non-empty, indexed set
of land trucks, LT;

type

1 CLI

2 TPS, TPs = TP-**set**, TP

3 VS, Vs = V-**set**, V

4 LTS, LTs = LT-**set**, LT

value

2 obs_TPS: CLI → TPS, obs_TPs: CLI → TPs

3 obs_VS: CLI → VS, obs_Vs: CLI → Vs

4 obs_LTS: TP → LTS, obs_LTs: LTS → LTs

axiom

2 ∀ cli:CLI·**card** obs_TPs(obs_TPS(cli))≥2

3 ∀ cli:CLI·**card** obs_Vs(obs_VS(cli))≥1

4 ∀ lts:LTS·**card** obs_LTs(obs_LTS(cli))≥1

5.1.1. Terminal Ports

In a terminal port one can observe

5 a [composite] container stowage area, CSA;

6 a non-empty, indexed set of one or more quay cranes, QC;

7 a non-empty, indexed set of one or more terminal cranes, TC;

8 a non-empty, indexed set of quay trucks, QT;

9 a[n atomic] quay[2], Q[3]; and

10 a[n atomic] terminal port monitoring and control center, MCC.

---

[2] We can, without loss of generality, describe a terminal as having exactly one quay (!) – just as we, again without any loss of generality, describe it as having exactly one container stowage area.

[3] *Quay:* a long structure, usually built of stone, where boats can be tied up to take on and off their goods.

Pronunciation: key.

Thesaurus: berth, jetty, key, landing, levy, slip, wharf

type
5 CSA
6 QCS QCs = QC-**set**, QC
7 TCS, TCs = TC-**set**, TC
8 QTS, QTs = QT-**set**, QT
9 Q
10 MCC
value
5 obs_CSA: TP → CSA
6 obs_QCS: TP → QCS, obs_QCs: QCS → QCs
7 obs_TCS: TP → TCS, obs_TCs: TCS → TCs
8 obs_QTS: TP → QTS, obs_QTs: QTS → QTs
9 obs_Q: TP → Q
10 obs_MCC: TP → MCC
axiom
6 ∀ tss:TCS•**card** obs_TCs(tcs)≥1
7 ∀ qs:QS•**card** obs_Qs(qs)≥1
8 ∀ qcs:QCS•**card** obs_QCs(qcs)≥1
10 ∀ qts:QTS•**card** obs_QTs(qts)≥1

5.1.2. Quays

- Although container terminal port quays
  - can be modelled as composite parts
  - we have chosen to describe them as atomic.
  - We shall subsequently endow the single terminal port quay with such attributes as quay segments, quay positions and berthing[4].

[4] Berth: Sufficient space for a vessel to maneuver; a space for a vessel to dock or anchor; (whether occupied by vessels or not). Berthing: To bring (a vessel) to a berth; to provide with a berth.

5.1.3. Container Stowage Areas: Bays, Rows and Stacks

11 From a container stowage area one can observe a non-empty indexed set of bays,

12 From a bay we can observe a non-empty indexed set of rows.

13 From a row we can observe a non-empty indexed set of stacks.

14 From a stack we can observe a possibly empty indexed set of containers.

type

11 BAYS, BAYs = BAY-**set**, BAY

12 ROWS, ROWs = ROW-**set**, ROW

13 STKS, STKs = STK-**set**, STK

14 CONS, CONs = CON-**set**, CON

value

11 obs_BAYS: CSA → BAYS, obs_BAYs: BAYS → BAYs

12 obs_ROWS: BAY → ROWS, obs_ROWs: ROWS → ROWs

13 obs_STKS: ROW → STKS, obs_STKs: STKS → STKs

14 obs_CONS: STK → CONS, obs_CONs: CONS → CONs

axiom

11 ∀ bays:BAYs · **card** bays > 0

12 ∀ rows:ROWs · **card** rows > 0

13 ∀ stks:STKs · **card** stks > 0

5.1.4. Vessels

From (or in) a vessel one can observe

15 [5] a container stowage area

16 and some other parts.

type

5 CSA

16 ...

value

5 obs_CSA: V $\rightarrow$ CSA

16 ...

5.1.5. Functions Concerning Container Stowage Areas

17 One can calculate

18 the set of all container storage areas:

19 of all terminal ports together with those

20 of all container lines.

```
value
17  cont_stow_areas: CLI → CSA-set
18  cont_stow_areas(cli) ≡
19      {obs_CSA(tp)|tp:TP•tp ∈ obs_TPs(obs_TPS(cli))}
20    ∪ {obs_CSA(cl)|cl:CL•cl ∈ obs_CLs(obs_CLS(cli))}
```

One can calculate the containers of

21 a stack,

22 a row,

23 a bay, and

24 a container stowage area.

value

```
21  extr_cons_stack: STK → CONs
21  extr_cons_stack(stk)≡obs_CONs(obs_CONS(stk))
22  extr_cons_row: ROW → CONs
22  extr_cons_row(row)≡{obs_CONs(obs_CONS(stk))|stk:STK•stk∈obs_STKs(obs_STKS(stk))}
23  extr_cons_bay: BAY → CONs
23  extr_cons_bay(bay)≡{obs_CONs(obs_CONS(row))|row:ROW•row∈obs_ROWs(obs_ROWS(bay))}
24  extr_cons_csa: CSA → CONs
24  extr_cons_csa(csa)≡{obs_CONs(obs_CONS(bay))|bay:BAY•bay∈obs_BAYs(obs_BAYS(csa))}
```

5.1.6. Axioms Concerning Container Stowage Areas

25 All rows contain different, i.e. distinct containers.

26 All bays contain different, i.e. distinct containers.

27 All container stowage areas contain different, i.e. distinct containers.

value

25 $\forall$ cli:CLI $\bullet$

25 $\quad\forall$ csa,csa$'$:CSA$\bullet\{$csa,csa$'\}\subseteq$cont_stow_areas(cli) $\bullet$

25 $\quad\quad\forall$ row,row$'$:ROW $\bullet$

25 $\quad\quad\quad\{$row,row$'\}\subseteq$obs_ROWs(obs_ROWS(csa))$\cup$obs_ROWs(obs_ROWS(csa$'$))$\Rightarrow$

25 $\quad\quad\quad$extr_cons_row(row) $\cap$ extr_cons_row(row$'$) $= \{\}\ \wedge$

26 $\quad\quad\forall$ bay,bay$'$:BAY $\bullet$

26 $\quad\quad\quad\{$bay,bay$'\}\subseteq$obs_ROWs(obs_ROWS(csa))$\cup$obs_ROWs(obs_ROWS(csa$'$))$\Rightarrow$

26 $\quad\quad\quad$extr_cons_bay(bay) $\cap$ extr_cons_bay(bay$'$) $= \{\}\ \wedge$

27 $\quad\quad\quad$extr_cons_csa(csa) $\cap$ extr_cons_csa(csa$'$) $= \{\}$

5.1.7. Stacks

- **An aside:**
  - We shall use the term ‘stack’ in two senses:
    - (i) as a component of container storage area bays; and
    - (ii) to refer to the collection of stacks in a bay of a terminal container storage area.

28 Stacks are created empty, and hence stacks can be *empty*.

29 One can *push* a *container* onto a *stack* and obtain a *non-empty stack*.

30 One can *pop* a *container* from a *non-epmpty stack* and obtain a pair of a *container* and a possibly empty *stack*.

value

28 empty: () $\rightarrow$ STK, is_empty: STK $\rightarrow$ **Bool**

29 push: CON $\times$ STK $\rightarrow$ STK

30 pop: STK $\xrightarrow{\sim}$ (CON $\times$ STK)

axiom

28 is_empty(empty()), $\sim$is_empty(push(c,stk))

29 pop(push(c,stk)) = (c,stk)

30 **pre** pop(stk),pop(push(c.stk)): $\sim$is_empty(stk)

30 pop(empty()) = **chaos**

5.2. Terminal Port Command Centers
5.2.1. Discussion

- We consider terminal port monitoring & control command centers to be atomic parts.
- The purpose of a terminal port command center is to monitor and control
  - the allocation and service (berthing) of any visiting vessel to quay positions and by quay cranes,
  - the allocation and services of vessels by quay cranes,
  - the allocation and services of quay cranes by quay trucks,
  - the allocation and services of quay trucks to quay cranes, containers and terminal stacks,
  - the allocation and services of land trucks to containers and terminal stacks,

- This implies that there are means for communication between a terminal command center and
  - vessels,
  - quay cranes,
  - stack cranes,
  - quay trucks,
  - land trucks,
  - terminal stacks and
  - containers.

5.2.2. Justification

- We shall justify the concept of terminal monitoring & control, i.e., command centers.
  - First we know that we are going, through a transcendental deduction, to model certain parts as behaviours.
  - These parts, we decide, after some analysis that we forego, to be
    - vessels,
    - quay cranes,
    - quay trucks,
    - stack cranes
    - stacks,
    - land trucks, and
    - containers.

- Behaviours are usually like **actors**: they can instigate actions.
- But we decide, in our analysis, that some of these behaviours,
  - quay cranes,
  - quay trucks,
  - stack cranes and
  - stacks,

  are *"passive" actors:*
  - are behaviourally not endowed
  - with being able to initiate "own" actions.

- Instead, therefore, of all these behaviours,
  - being able to communicate directly, pairwise,
  - as loosely indicated by the figures of Pages 43 and 49,
  - we model them to communicate *via* their terminal command centers.

- *This is how we justify the introduction of the concept of terminal command centers.*

- They are an abstraction.
  - In *"ye olde days"*
    - you could observe, not one, but, perhaps, a hierarchy
    - of terminal port offices, staffed by people,
    - [each office, each group of staff] with its set of duties:
      - communicating (by radio-phone) with approaching [and departing] vessels;
      - scheduling quay positions, quay cranes and quay trucks;
      - managing the operation of cranes and trucks;
      - and, on a large scale,
      - calculating stowage: on vessels and in terminals.

Today, *"an age of ubiquitous computing"*,

- most of these offices and their staff are replaced by
- electronics: sensors, actuators, communication and computing,
- and with massive stowage data processing:
  - where should containers be stowed
  - on board vessels and in terminals
  - so as to near-optimise all operations.

5.3. Unique Identifications

31 Vessels have unique identifiers, VI.

32 Quay cranes have unique identifiers.

33 Quay trucks have unique identifiers.

34 Stack cranes have unique identifiers.

35 Bays (“Stacks”) of terminal container stowage areas have unique identifiers, TBI, a proper subset of BI.

36 Land trucks have unique identifiers, LTI.

37 Terminal port command centers have unique identifiers, MCCI.

38 Containers have unique identifiers, CI.

39 Bays of container stowage areas have unique identifiers, BI.

40 Rows of a bay have unique identifiers, RI.

41 Stacks of a row have unique identifiers, SI.

42 The part unique identifier types are mutually disjoint.

type

31 VI
32 QCI
33 QTI
34 SCI
35 TBI
36 LTI
37 MCCI
38 CI
39 BI
40 RI
41 SI

value

31 uid_V: V → VI
32 uid_QC: QC → QCI
33 uid_QT: QT → QTI
34 uid_SC: SC → SCI
34 uid_TBI: BAY → TBI
35 uid_LT: LT → LTI
37 uid_MCC: MCC → MCCI
37 uid_CON: CON → CI
34 uid_BAY: BAY → BI
35 uid_ROW: ROW → RI
36 uid_STK: STK → SI

axiom

42 VI, QCI, QTI, SCI, TBI, LTI, MCCI, CI, RI and SI mutually disjoint
42 TBI ⊂ BI

5.3.1. Unique Identifiers: Distinctness of Parts

43 If two containers are different
then their unique identifiers must be different.

axiom

43 $\forall$ con,con′:CON · con $\neq$ con′ $\Rightarrow$ uid_CON(con) $\neq$ uid_CON(con′)

- The same distinctness criterion applies to
  - stacks,
  - rows,
  - bays,
  - container storage areas,
  - terminal ports,
  - cranes,
  - vessels,
  - etc.

5.3.2. Unique Identifiers: Two Useful Abbreviations

Container positions within a container stowage area can be represented in two ways:

44 by a triple of a bay identifier, a row identifier and a stack identifier, and

45 by these three elements and a tier position (i.e., position within a stack).

44 $\mathsf{BRS} = \mathsf{BI} \times \mathsf{RI} \times \mathsf{SI}$

45 $\mathsf{BRSP} = \mathsf{BI} \times \mathsf{RI} \times \mathsf{SI} \times \textbf{Nat}$

axiom

45 $\forall\ (\mathsf{bu,ri,si,n}):\mathsf{BRSP} \cdot \mathsf{n}>0$

5.3.3. Unique Identifiers: Some Useful Index Set Selection Functions

46 From a container stowage area once can observe all bay identifiers.

47 From a bay once can observe all row identifiers.

48 From a row once can observe all stack identifiers.

49 From a virtual container storage area, i.e., an icsa:iCSA, one can extract all the unique container identifiers.

value

```
46  xtr_BIs: CSA → BI-set
46  xtr_BIs(csa) ≡ {uid_BAY(bay)|bay:BAY•bay ∈ xtr_BAYs(csa)}

46  xtr_RIs: BAY → RI-set
47  xtr_RIs(bay) ≡ {uid_ROW(bay)|row:ROW•row ∈ obs_ROWs(bay)}

46  xtr_SIs: ROW → SI-set
48  xtr_SIs(row) ≡ {uid_STK(row)|stk:STK•stk ∈ obs_STKs(row)}

49  xtr_CIs: iCSA → CI-set
49  xtr_CIs(icsa) ≡
49      ... [to come] ...
```

5.3.4. Unique Identifiers: Ordering of Bays, Rows and Stacks

- The bays of a container stowage area are usually ordered.
  - So are the rows of bays,
  - and stacks of rows.
- Ordering is here treated as ***attributes*** of
  - container stowage areas,
  - bays and
  - stacks.
- We shall treat ***attributes*** further on.

5.4. States, Global Values and Constraints

5.4.1. States

50 We postulate a container line industry *cli*:CLI.

From that we observe, successively, all parts:

51 the set, *cs*:C-**set**, of all containers;

52 the set, *tps*:TPs, of all terminal ports;

53 the set, *vs*:Vs, of all vessels; and

54 the set, *lts*:LTs, of all land trucks.

value

50 *cli*:CLI
51 *cs*:C-**set** = obs_Cs(obs_CS(*cli*))
52 *tps*:TP-**set** = obs_TPs(obs_TPS(*cli*))
53 *vs*:V-**set** = obs_Vs(obs_VS(*cli*))
54 *lts*:LTs = obs_LTs(obs_LTS(*cli*))

We can observe

55 $csas$:CSA-**set**, the set of all terminal port container stowage areas of all terminal ports;

56 $bays$:BAY-**set**, the terminal port bays of all terminals;

57 the set, qcs:QC-**set**, of all quay cranes of all terminals;

58 the set, qts:QT-**set**, of all quay trucks of all terminal ports; and

59 the set, scs:SC-**set**, of all terminal (i.e., stack) cranes of all terminal ports.

value

55 $csas$:CSA-**set** = {obs_CSA(tp)|tp:TP•tp ∈ tps}
55 $bays$:BAY-**set** = {obs_BAY(csa)|csa:CSA•csa ∈ $csas$}
57 qcs:QC-**set** = {obs_QCs(obs_QCS(tp))|tp:TP•tp ∈ tps}
58 qts:QT-**set** = {obs_QTs(obs_QTS(tp))|tp:TP•tp ∈ tps}
59 scs:SC-**set** = {obs_SCs(obs_SCS(tp))|tp:TP•tp ∈ tps}

5.4.2. Unique Identifiers

Given the generic parts outlined on Slide 77 we can similarly define generic sets of unique identifiers.

60 There is the set, c_uis, of all container identifiers;

61 the set, tp_uis, of all terminal port identifiers;

62 the set, mcc_uis, of all terminal port command center identifiers;

63 the set, v_uis, of all vessel identifiers;

64 the set, qc_uis, of quay crane identifiers of all terminal ports;

65 the set, qt_uis, of quay truck identifiers of all terminal ports;

66 the set, sc_uis, of stack crane identifiers of all terminal ports;

67 the set, stk_uis, of stack identifiers of all terminal ports;

68 the set, lt_uis, of all land truck identifiers; and

69 the set, uis, of all vessel, crane and truck identifiers.

value

60 c_uis:CI-**set** = {uid_C(c)|c:C•c∈cs}
61 tp_uis:TPI-**set** = {uid_TP(tp)|tp:TP•tp∈tps}
62 mcc_uis:TPI-**set** = {uid_MCC(obs_MCC(tp))|tp:TP•tp∈tps}
63 v_uis:VI-**set** = {uid_V(v)|v:V•v∈vs}
64 qc_uis:QCI-**set** = {uid_QC(qc)|qc:QC•qc∈qcs}
65 qt_uis:QTI-**set** = {uid_QT(qt)|qt:QT•qt∈qts}
66 sc_uis:SCI-**set** = {uid_SC(sc)|sc:SC•sc∈scs}
67 stk_uis:BI-**set** = {uid_BAY(stk)|stk:BAY•stk∈$stks$}
68 lt_uis:LTI-**set** = {uid_LL(lt)|lt:LT•lt∈lts}
69 uis:(VI|QCI|QTI|SCI|BI|LTI)-**set** = $v\_uis \cup qc\_uis \cup qt\_uis \cup sc\_ui$

70 the map, *tpmcc_idm*, from terminal port identifiers into the identifiers of respective command centers;

71 the map, *mccqc_idsm*, from command center identifiers into the set of quay crane identifiers of respective ports;

72 the map, *mccqt_idsm*, from command center identifiers into the identifiers of quay trucks of respective ports;

73 the map, *mccsc_idsm*, from command center identifiers into the identifiers of quay trucks of respective ports; and

74 the map, *mccbays_idsm*, from command center identifiers
into the set of bay identifiers (i.e., "stacks") of respective ports;

value

70 $tpmcc\_idm$:(TI $\overrightarrow{m}$ MCCI) = [uid_TP(tp)$\mapsto$uid_MCC(obs_MCC(tp))|tp:TP•tp $\in$ tps]
71 $mccqc\_idsm$:(MCCI $\overrightarrow{m}$ QCI-**set**)
71 = [$tpmcc\_uim$(uid_TP(tp)) $\mapsto$ { uid_QC(qc)
71 | qc:QC • qc $\in$ obs_QCs(obs_QCS(tp)) } | tp:TP•tp $\in$ tps]
72 $mccqt\_idsm$:(MCCI $\overrightarrow{m}$ QTI-**set**) =
72 = [$tpmcc\_uim$(uid_TP(tp)) $\mapsto$ { uid_QT(qt)
72 | qt:QT • qt $\in$ obs_QTs(obs_QTS(tp)) } | tp:TP•tp $\in$ tps]
73 $mccsc\_idsm$:(MCCI $\overrightarrow{m}$ SCI-**set**)
73 = [$tpmcc\_uim$(uid_TP(tp)) $\mapsto$ { uid_SC(sc)
73 | sc:SC • sc $\in$ obs_SCs(obs_SCS(tp)) } | tp:TP•tp $\in$ tps]
74 $mccbays\_idsm$:(MCCI $\overrightarrow{m}$ BI-**set**)
74 = [$tpmcc\_uim$(uid_TP(tp)) $\mapsto$ { uid_B(b)
74 | b:BAY•b $\in$ obs_BAYs(obs_BAYS(obs_CSA(tp)))} | tp:TP•tp $\in$ tps]

5.4.3. Some Axioms on Uniqueness

TO BE WRITTEN

5.5. Mereology

5.5.1. Physical versus Conceptual Mereology

- We briefly discuss a distinction that was not made in [13]:
  - whether to base a mereology on *physical connections*
  - or on *functional* or, as we shall call it, *conceptual relations*.
- We shall, for this domain model, choose the conceptual view.
  - The physical mereology view can be motivated, i.e. justified, from the figures on pages 43 and 49.
  - The conceptual view is chosen on the basis of the justification of the terminal command centers.
  - We shall model physical mereology as attributes.[5]

[5]Editorial note: Names of physical and of conceptual mereologies have to be "streamlined". As now, they are a "mess" !

5.5.2. Vessels

5.5.2.1 Physical Mereology:

75 Vessels are physically “connectable” to quay cranes of any terminal port.

```
type
75  Phys_V_Mer = QCI-set
value
75  attr_Phys_V_Mer: V → Phys_V_mer
```

5.5.2.2 Conceptual Mereology:

76 Container vessels can potentially visit any container terminal port, hence have as [part of] their mereology, a set of terminal port command center identifiers.

type

76 V_Mer = MCCI-**set**

value

76 mereo_V: V → V_Mer

axiom

76 ∀ v:V · v ∈ vs ⇒ mereo_V(v) ⊆ mcc_uis

5.5.3. Quay Cranes

5.5.3.1 Physical Mereology:

- In modelling the physical mereology,
- though as an attribute, of quay cranes,
- we need the notion of quay positions.

77 Quay cranes are, at any time, positioned at one or more adjacent quay positions of an identified segment of such.

type

77 Phys_QC_Mereo = QPSId $\times$ QP*

value

77 attr_Phys_QC: QC $\rightarrow$ Phys_QC_Mereo

78 The quay positions, **qcmereo = (qpsid,qpl):QCMereo**, must be proper quay positions of the terminal,

79 that is, the segment identifier, **qpsid**, must be one of the terminal,

80 and the list, **qpl**, must be contiguously contained within the so identifier segment.

axiom ∀ tp:TP,
78 **let** q = obs_Q(tp), qcs = obs_QCs(obs_QCS(tp)) **in**
79 ∀ q:Q · q ∈ qcs ⇒
79 **let** (qpsid,qpl) = obs_Mereo(q), qps = attr_QPSs(q) **in**
79 qpsid ∈ **dom** qps
80 ∧ ∃ i,j:**Nat** · {i,j} ∈ **inds** qpl ∧ ⟨(qps(qpsi))[k]|i≤k≤j⟩ = qpl
78 **end end**

5.5.3.2 Conceptual Mereology:

- The conceptual mereology is simpler.

81 Quay cranes are conceptually related to the command center of the terminal in which they are located.

type

81 QC_Mer = MCCI

value

81 mereo_QC: QC → QC_Mer

5.5.4. Quay Trucks

5.5.4.1 Physical Mereology:

82 Quay trucks are physically "connectable" to quay and stack cranes.

type
82 Phys_QT_Mer = QCI-**set** × QCI-**set**
value
82 attr_Phys_QT_Mer: QT → Phys_QT_Mer

5.5.4.2 Conceptual Mereology:

83 Quay trucks are conceptually connected to the command center of the terminal port of which they are a part.

type
83 QT_Mer = MCCI
value
83 mereo_QT: QT → QT_Mer

5.5.5. Stack Cranes
5.5.5.1 Physical Mereology:

84 Terminal stack cranes are positioned to serve one or more terminal area bays, one or more quay trucks and one or more land trucks.

85 The terminal stack crane positions are indeed positions of their terminal

86 and no two of them share bays.

type
84 Phys_SCmereo = s_bis:BI-**set** × s_qtis:QTI-**set** × s_ltis:LTI-**set**
axiom
84 ∀ (bis,qtis,ltis):Phys_SCmereo•bis≠{} ∧ qtis≠{} ∧ ltis≠{}
value
84 Phys_SCmereo: SC → Phys_SCmereo
axiom
84 ∀ tp:TP •
84 **let** csa=obs_CSA(tp), bays=obs_BAYs(obs_BAYS(csa)), scs=obs_SCs(obs_SCS(tp)) **in**
85 ∀ sc:SC•sc ∈ scs ⇒ Phys_SCmereo(sc) ⊆ xtr_BIs(csa)
86 ∧ ∀ tp′,tp″:TP•{tc′,tc″}⊆tcs ∧ tc′≠tc″
86 ⇒ s_bis(Phys_SCmereo(tc′)) ∩ s_bis(Phys_SCmereo(tc″))={} **end**

5.5.5.2 Conceptual Mereology:

- The conceptual stack crane mereology is simple:

87 Each stack is conceptually related to the command center of the terminal at which it is located.

type
87 SC_Mer = MCCI
value
87 mereo_SC: SC → SC_Mer

5.5.6. Container Stowage Areas

5.5.6.1 Bays, Rows and Stacks:

The following are some comments related to, but not defining a mereology for container stowage areas.

88 A bay of a container stowage area

a. has either a predecessor
b. or a successor,
c. or both (and then distinct).
d. No row cannot have neither a predecessor nor a successor.

89 A row of a bay has a predecessor and a successor, the first stack has no predecessor and the last stack has no successor.

90 A stack of a row has a predecessor and a successor, the first stack has no predecessor, and the last stack has no successor.

```
value
88   BAY_Mer: BAY → ({|'nil'|}|BI) × (BI|{|'nil'|})
89   ROW_Mer: ROW → ({|'nil'|}|RI) × (RI|{|'nil'|})
90   STK_Mer: STK → ({|'nil'|}|SI) × (SI|{|'nil'|})
axiom
88   ∀ csa:CSA • let bs = obs_BAYs(obs_BAYS(csa)) in
88       ∀ b:BAY • b ∈ bs ⇒
88           let (nb,nb′) = mereo_BAY(b) in
88           case (nb,nb′) of
88a.             ('nil',bi) → bi ∈ xtr_BIs(csa),
88b.             (bi,'nil') → bi ∈ xtr_BIs(csa),
88d.             ('nil','nil') → chaos,
88c.             (bi,bi′) → {bi,bi′} ⊆ xtr_BIs(csa) ∧ bi≠bi′
88           end end end
89    as for rows
90    as for stacks
```

5.5.7. Bay Mereology

5.5.7.1 Physical Vessel Bay Mereology:

91 A vessel bay is topologically related to the vessel on board of which it is placed and to the set of all quay cranes of all terminal ports.

type
91 Phys_VES_BAY_Mer = VI × QCI-**set**

5.5.7.2 Conceptual Vessel Bay Mereology:

92 A vessel bay is conceptually related to the set of all command centers of all terminal ports.

type
92 V_BAY_Mer = MCCI-**set**

5.5.7.3 Physical Terminal Port Bay (cum Stack) Mereology:

93 A terminal bay (cum stack) is topologically related to the stack cranes of a given terminal port and all land trucks.

type

93 $\text{Phys\_STK\_Mer} = \text{SCI-}\mathbf{set} \times \text{LTI-}\mathbf{set}$

5.5.7.4 Conceptual Terminal Port Bay (cum Stack) Mereology:

94 A terminal port bay is conceptually related to the command center of its port.

type

94 $\text{T\_BAY\_Mer} = \text{MCCI}$

5.5.8. Land Trucks

5.5.8.1 Physical Mereology:

95 Land trucks are physically "connectable" to stack cranes – of any port.

type
95 Phys_LT_Mer = SCI-**set**
value
95 attr_Phys_LT_Mer: LT → Phys_LT_Mer

5.5.8.2 Conceptual Mereology:

96 Land trucks are conceptually connected to the command centers of any terminal port.

type
96 LT_Mer = MCCI-**set**
value
96 mereo_LT: LT → LT_Mer

5.5.9. Command Center

- Command centers are basically conceptual quantities.
  - ◈ Hence we can expect the physical mereology
  - ◈ to be the conceptual mereology.

97 . Command centers are physically and conceptually connected to

- all vessels,
- all cranes of the terminal port of the command center,
- all quay trucks of the terminal port of the command center,
- all stacks (i.e., bays) of the terminal port of the command center, and
- all land trucks, and
- all containers.

```
type
97  MCC_Mer = VI-set×QCI-set×QTI-set×SCI-set×BI-set×LTI-set×CI-set
value
97  mereo_MCC: MCC → MCC_Mer
axiom
97  ∀ tp:TP • tp ∈ tps •
97     let qcs:QC-set • qcs = obs_QCs(obs_QCS(tp)),
97         qts:QT-set • qts = obs_QTs(obs_QTS(tp)),
97         scs:SC-set • scs = obs_SCs(obs_SCS(tp)),
97         bs:iBAY-set • bs = obs_Bs(obs_BS(obs_CSA(tp))) in
97     let vis:VI_set • vis = {uid_VI(v)|v:V•v ∈ vs},
97         qcis:QCI_set • qcis = {uid_QCI(qc)|qc:QC•qc ∈ qcs},
97         qtis:QTI_set • qcis = {uid_QTI(qc)|qt:QT•qt ∈ qts},
97         scis:SCI-set • scis = {uid_SCI(sc)|sc:SC•sc ∈ scs},
97         bis:iBAY-set • bis = {uid_BI(b)|b:iBAY•b ∈ bs},
97         ltis:LTI-set • ltis = {uid_LTI(lt)|lt:LT•lt ∈ lts},
97         cis:SCI-set • cis = {uid_CI(c)|c:C•c ∈ cs} in
97     mereo_MCC(obs_MCC(tp)) = (vis,qcis,scis,sis,bis,ltis,cis) end end
```

5.5.10. Conceptual Mereology of Containers

- The physical mereology of any container is modelled as a container attribute.

98 The conceptual mereology is modelled by containers being connected to all terminal command centers.

type

98 C_Mer = MCCI-**set**

value

98 mereo_C: C → C_Mer

axiom

98 ∀ c:C · mereo_C(c) = mcc_uis