Rivers and Canals – Endurants
A Technical Note
Incomplete Draft

Dines Bjørner
Prof. Emeritus: Technical University of Denmark
Private: Fredsvej 11, DK-2840 Holte, Danmark
E–Mail: bjorner@gmail.com, Home Page: www.imm.dtu.dk/~db

March 12, 2021, 13:21

Abstract

Presently this document represents a technical-scientific note. It is technical in that much of the material can be found in other technical notes of mine. It is – perhaps – scientific in that I am searching for a nice, well, beautiful, way of modeling canals, such as for example those of the Dutch Rijkswaterstaat[1]. I am fascinated with Holland's tackling of their land/water/river/ocean levels.

I am presently, March 12, 2021, 13:21, working on Sect. 4.3.4.2 on page 46.

Contents

[1] https://www.rijkswaterstaat.nl/ The Dutch canal system is first and foremost, it appears, for the control of water levels, secondly for ship/barge/boat navigation.

1 Introduction

1.1 Waterways

By waterways we mean rivers, canals, lakes and oceans – such as are navigable by vessels: barges, boats and ships.

• • •

Disclaimer: At present ("great") lakes and the oceans (there are two !) are not included in this modeling effort.

1.1.1 General

Canals are artificial or human-made channels or waterways that are used for navigation, transporting water, crop irrigation, or drainage purposes. Therefore, a canal can be considered an artificial version of a river. Canals are artificial or human-made channels or waterways that are used for navigation, transporting water, crop irrigation, or drainage purposes. Therefore, a canal can be considered an artificial version of a river.

Rivers, on the other hand, are naturally flowing watercourses, and typically flow until discharging their water into a lake, sea, ocean, or another river, while canals are constructed to connect existing rivers, seas, or lakes. However, occasionally some rivers do not discharge their water into lakes, seas, oceans, or other rivers. Rivers that do not empty into another body of water might flow into the ground or simply dry up before reaching another body of water. Additionally, small rivers can also be referred to as streams, rivulets, creeks, rills, or brooks.

The natural water system of the earth includes 71% ocean with land continents being traversed by brooks, rivers, lakes and river deltas.

Headwaters are streams and rivers (tributaries) that are the source of a stream or river.

A tributary is a river or stream that flows into another stream, river, or lake.

A delta is a large, silty area at the mouth of a river at which the river splits into many different slow-flowing channels that have muddy banks. New land is created at deltas. Deltas are often triangular-shaped, hence the name (the Greek letter 'delta' is shaped like a triangle).

The trunk is the main course of river.

Confluence: In geography, a confluence (also: conflux) occurs where two or more flowing bodies of water join together to form a single flow. A confluence can occur in several configurations: at the point where a tributary joins a larger river (main stem); or where two streams meet to become the source of a river of a new name; or where two separated channels of a river (forming a river island) rejoin at the downstream end.

Towns and Harbours: In this report we model towns. That is, we therefore also model that towns have harbours – allowing river (and canal) vessels to berth (a place for mooring in a harbour) for cargo loading, unloading and resting.

1.1.2 Visualisation of Rivers and Canals

1.1.2.1 Rivers: Figures 1 and 2 illustrate a number of rivers.

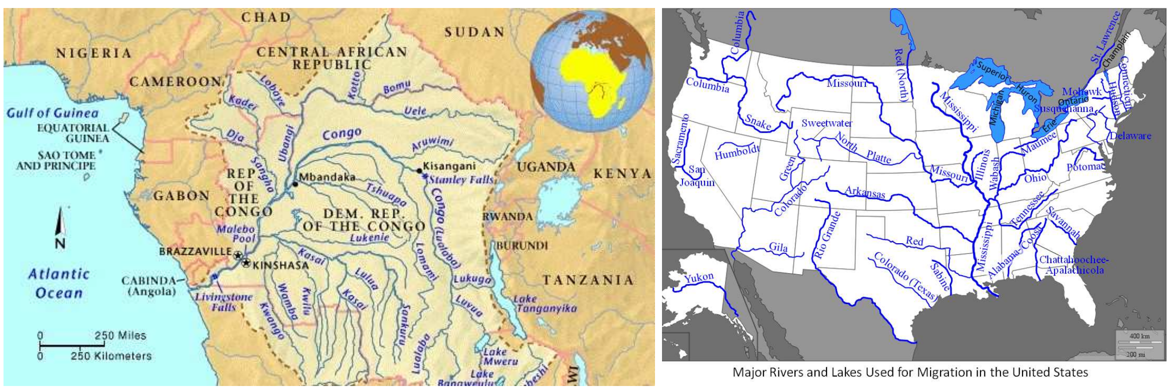

Figure 1: The Congo and the US Rivers

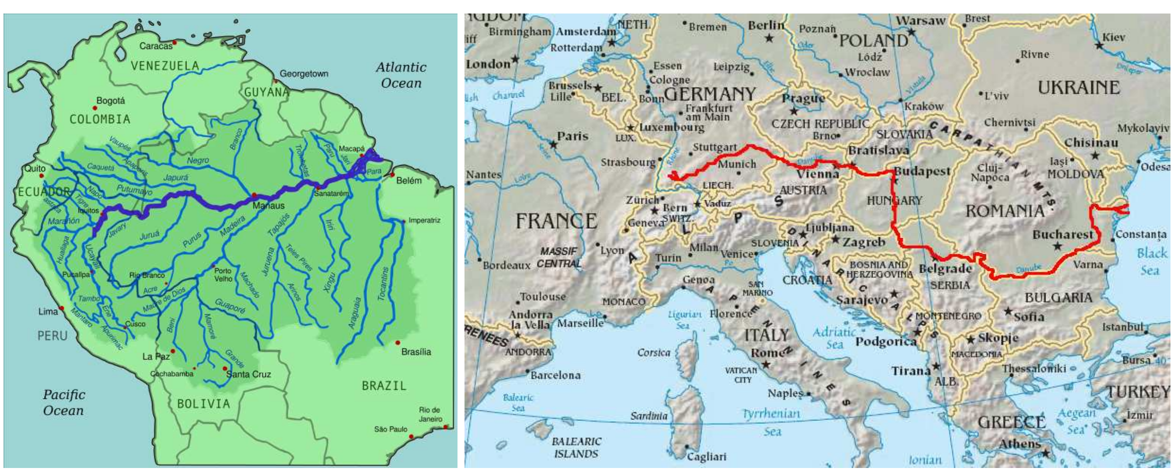

Figure 2: The Amazon and The Danube Rivers

1.1.2.2 Deltas: We illustrate four deltas, Fig. 3:

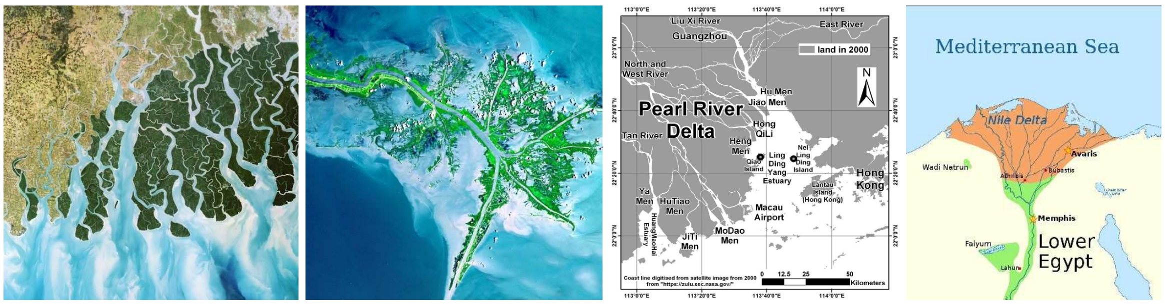

Figure 3: The Ganges, Mississippi, Pearl and the Nile Deltas

1.1.2.3 Canals and Water Systems: We illustrate just four ship/barge/boat and water level control canal systems, Figs. 4, 5, 6 on page 8 and 7 on page 9.

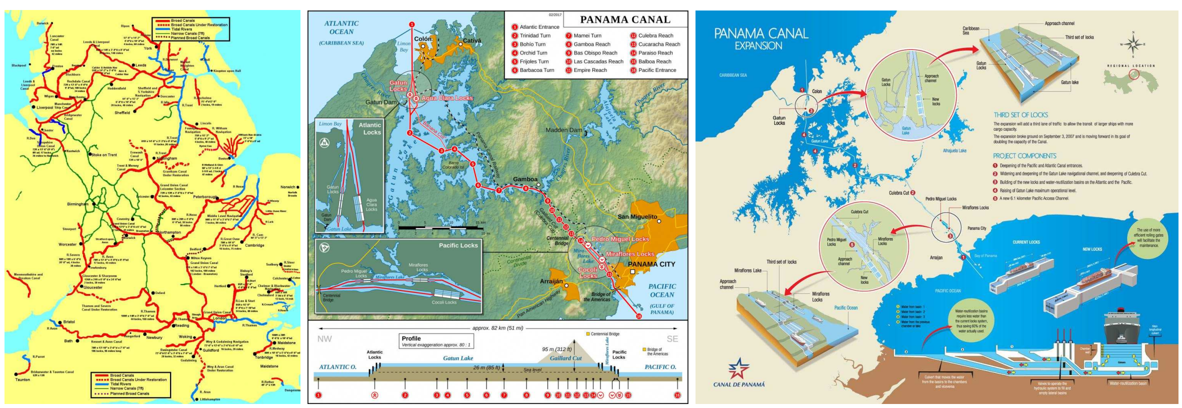

Figure 4: UK Canals and The Panama Canal

The rightmost figure of Fig. 7 is from the Dutch *Rijkswaaterstaat:* www.rijkswaterstaat.nl/english/.

1.1.2.4 Locks: A lock is a device used for raising and lowering boats, ships and other watercraft between stretches of water of different levels on river and canal waterways. The distinguishing feature of a lock is a fixed chamber in which the water level can be varied. Locks are used to make a river more easily navigable, or to allow a canal to cross land that is not level. Later canals used more and larger locks to allow a more direct route to be taken.[2]

We illustrate a number of locks: Figs. 8 on page 10 and 9 on page 10.

[2] https://en.wikipedia.org/wiki/Lock_(water_navigation)

Figure 5: The Swedish Göta Kanal

1.2 Structure of This Report

Rivers are narrated and formalised in Sects.:

- 2 [Endurants],
- 3.1 [Unique Identifiers],
- 3.2 [Mereology],
- 3.4 [Attributes] and
- A.1 [Terminology].

2 Rivers: External Qualities – The Endurants

1. A river net is modeled as a graph, more specifically as a tree. The *root* of that river net tree is the mouth (or delta) of the river net. The *leaves* of that river net tree are the sources of respective trees. Paths from leaves to the root define *flows* of water.

2. We can thus, from a river net observe vertices

3. and edges.

4. River vertices model either a *source:* **so:SO**, a *mouth:* **mo:MO**, or possibly some *confluence:* **ko:KO**.

   A river may thus be "punctuated" by zero or more confluences, k:KO.

   A confluence defines the joining a 'main' river with zero[3] or more rivers into that 'main' river.

   We can talk about the "upstream" and the "downstream" of rivers from their confluence.

5. River edges model *stretches:* st:ST.

   A stretch is a linear sequences of simple, se:SE, or composite ce:CE, river elements.

6. River elements are either simple: (ch) river channels, which we shall call *river channels*: **CH**, or (la) lakes: **LA**, or (lo) locks: **LO**, or (wa) waterfalls (or *rapids*): **WA**, or (da) dams: **DA**, or

[3]Normally, though, one would expect, not zero, but one

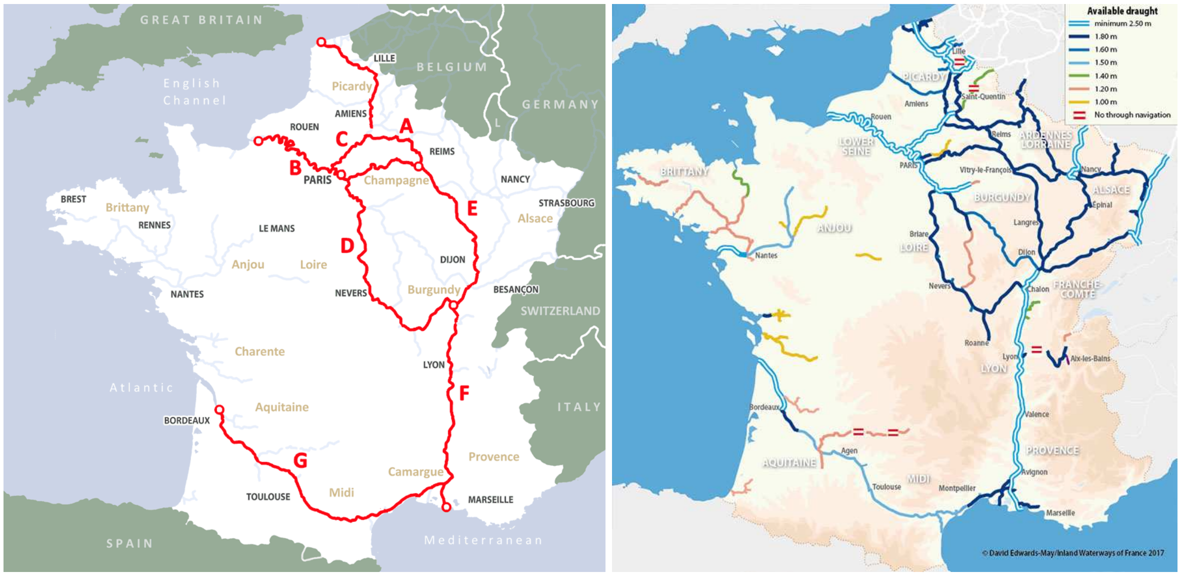

Figure 6: French Rivers and Canals

(to) towns (cities, villages): **to:TO**[4]; or composite, ce:CE: a dam with a lock, **(da:DA,la:LA)**, a town with a lake, **(to:TO,la:LA)**, etcetera; even a town with a lake and a confluence, **to:TO,-la:LA,ko:KO**. Etcetera.

type
1. RiN
2. V
3. E
4. SO, MO, KO
5. ST = (SE|CE)*
6. CH, LA, LO, WA, KO, DA, TO
6. SE = CH | LA | LO | WA | DA | TO
6. DaLo, WaLo, ToLa, ToLaKo, ...
6. CE = DaLo | WaLo | ToLa | ToLaKo | ...

value
4. obs_Vs: RiN $\to$ V-**set**
4. **axiom**
4. $\forall$ g:G,vs:V-**set**•vs $\in$ obs_Vs(g) $\Rightarrow$ vs$\neq${}
4. $\wedge$ $\forall$ v:V•v $\in$ vs $\Rightarrow$ is_SO(v) $\vee$ is_KO(v) $\vee$ is_MO(v)
5. obs_Es: RiN $\to$ E-**set**
5. **axiom**
5. $\forall$ g:G,es:E-**set**•es $\in$ obs_Es(g) $\Rightarrow$ es$\neq${}
5. $\wedge$ $\forall$ e:E•e $\in$ es $\Rightarrow$ is_ST(e)

[4] Towns is here really a synonym for river harbours, places along the river (or a canal) where river vessels can stop (moor) for the loading and unloading of cargo and for resting.

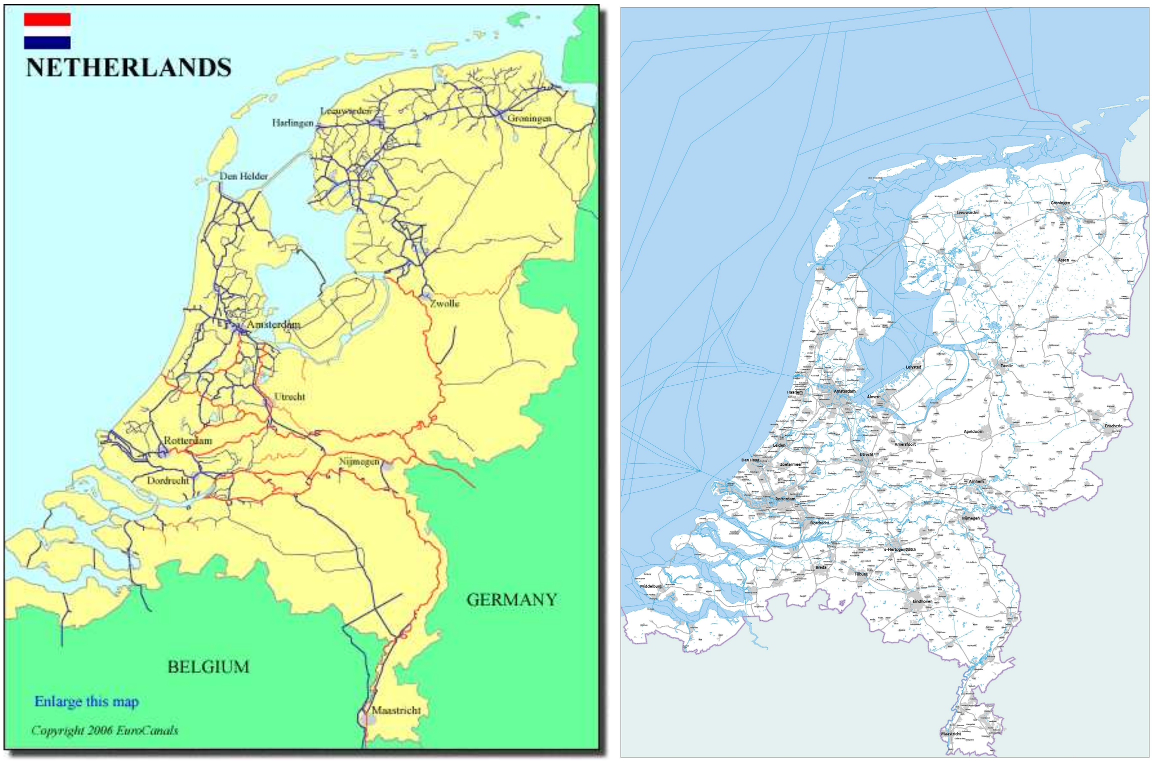

Figure 7: Dutch Rivers and Canals

5. obs_ST: E → ST
1. xtr_In_Degree_0_Vertices: RiN → SO-**set**
1. xtr_Out_Degree_0_Vertex: RiN → MO

3 Rivers: Internal Qualities

We refer to [1, Chapter 5]

3.1 Rivers: Unique Identifiers

We shall associate unique identifiers both with vertices, edges and verteax and edge elements.

7. River net vertices and edges have unique identifiers.
8. River net sources, confluences and mouths have unique identifiers.
9. River net stretches have unique identifiers.
10. River net channels, lakes, locks, waterfalls, dams and towns as well as combinations of these, that is, simple and composite river entities have unique identifiers.

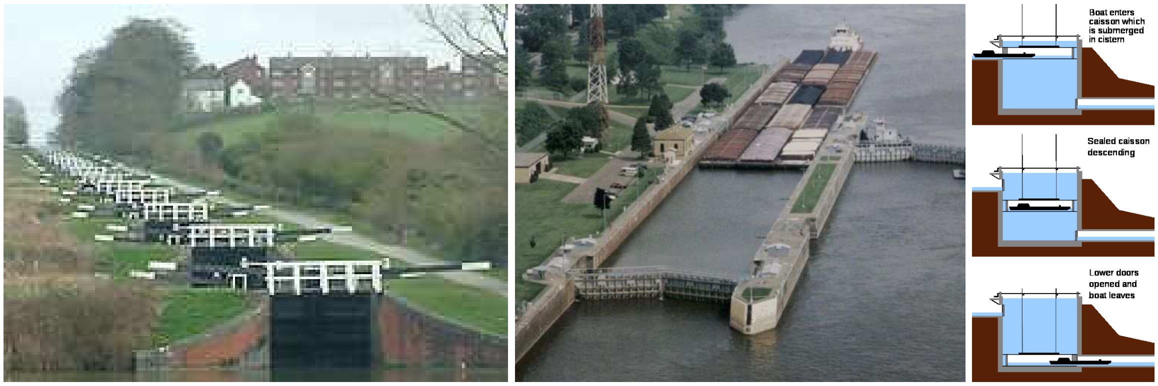

Figure 8: Inland Canal Locks

Figure 9: Harbour Canal Locks

type
7. V_UI, E_UI
8. SO_UI, KO_UI, MO_UI
9. ST_UI
10. CH_UI, LA_UI, LO_UI, WA_UI, DA_UI, TO_UI, DaLo_UI, WaLo_UI, ToLa_UI, ToLaKo_UI, ...

value
7. uid_V: V→V_UI, uid_E: E→ E_U
8. uid_SO: SO→SO_UI, uid_KO: KO→KO_UI, uid_MO: MO→MO_UI,
9. uid_ST: ST→ST_UI
10. uid_CH: CH→CH_UI, uid_LA: LA→LA_UI, uid_LO: LO→LO_UI, uid_WA: WA→WA_UI,
10. uid_DA: DA→DA_UI, uid_TO: TO→TO_UI,
10. uid_DaLo: DaLo→DaLo_UI, uid_WaLo: WaLo→WaLo_UI, uid_ToLa: ToLa→ToLa_UI,
10. uid_ToLaKo: ToLaKo→ToLaKo_UI, ...

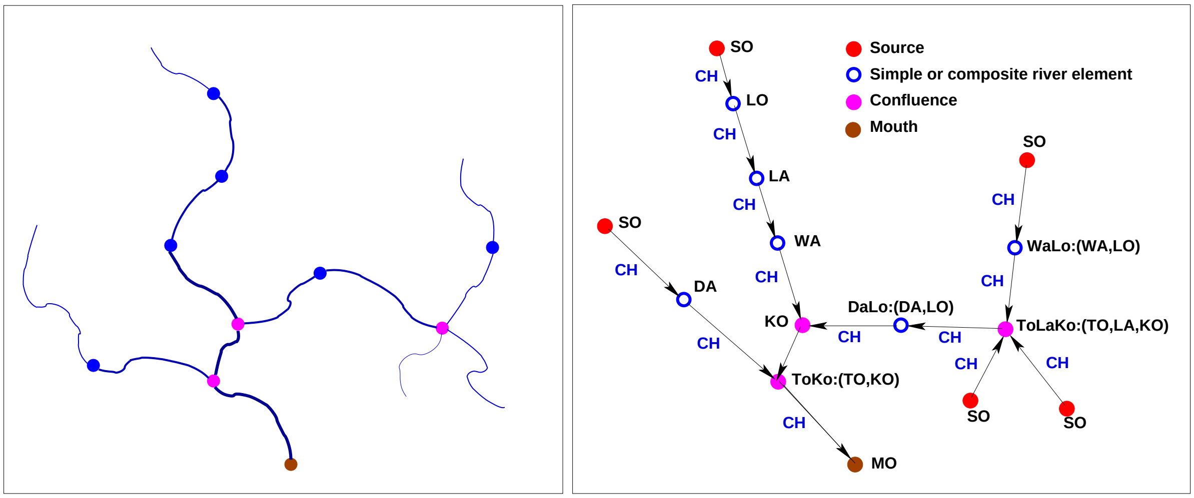

Figure 10: The “Composition” of a River Net: Right Tree is an abstraction of the Left Tree

11. All these identifiers are distinct.

The ⋒ operator takes the pairwise intersection of the types in its argument list and examines them for disjointness.

axiom

11. ⋒(V_UI,E_UI,SO_UI,KO_UI,MO_UI,ST_UI,CH_UI,

11.   LA_UI,LO_UI,WA_UI,DA_UI,TO_UI,DaLo_UI,WaLo_UI,ToLa_UI,ToLaKo_UI)

12. There are [many] other constraints, please state them !

12. [left as exercise to the reader !]

3.2 Rivers: Mereologies

13. The mereology of a river vertex is a pair: a set of unique identifiers, E_UI, of river edges, i.e., stretches, linear sequences of simple and composite river elements, incident upon the vertex, and a set of unique identifiers, E_UI, of river edges emanating from the vertex. If the vertex is a source then the first element of this pair is empty. If the vertex is a mount then the second element of this pair is empty. For a confluence vertex both elements of the pair are non-empty.

14. The mereology of a river edge, that is, the linear sequence of simple and composite river elements between two adjacent vertices, is a pair: the first element is a unique identifier of a river vertex and so is the second element of the pair.

We present the river net mereology in two forms. The first was with respect to its graph rendition. The second is with respect to its river element rendition.

15. The mereology of a source is just the single unique identifier of the first simple or composite river element of the stretch emanating from the source.

16. The mereology of a confluence is a triplet: the single unique identifier of the last simple or composite river element of the stretch of the main river incident upon the source, a set of unique identifier of the last simple or composite river element of the stretches of the tributary rivers incident upon the source, and the single unique identifier of the first simple or composite river element of the main river stretch emanating from the confluence.

17. The mereology of a mouth is just the single unique identifier of the last simple or composite river element of the stretch incident upon the mouth

18. The mereologies of simple and composite river elements are pairs: of the unique identifier of the river elements, including sources and confluences, upstream adjacent to the river element being "mereologised", and of the unique identifier of the river elements, including confluences and mouths, downstream adjacent to the river element being "mereologised".

13. Mer_V = E_UI-**set** × E_UI-**set**
14. Mer_E = V_UI × V_UI
15. Mer_SO = SE_UI | CE_UI
16. Mer_KO = (SE_UI|CE_UI) × (SE_UI|CE_UI)-**set** × (SE_UI|CE_UI)
17. Mer_MO = SE_UI | CE_UI
18. Mer_RE = (SO_UI|CO_UI|SE_UI|CE_UI) × (SE_UI|CE_UI|CO_UI|MO_UI)

19. The unique vertex and edge identifiers must be identifiers of the vertices and edges of a graph.

20. Similarly, the unique source, confluence and mouth identifiers must be identifiers of respective sources, confluences and mouths of a graph.

21. And likewise for simple and composite element identifiers.

22. No two sources, confluences, mouths, simple and composite elements have identical uinque identifiers.

23. There are other constraints, please state them !

axiom
19. [left as exercise to the reader !]
20. [left as exercise to the reader !]
21. [left as exercise to the reader !]
22. [left as exercise to the reader !]
23. [left as exercise to the reader !]

3.3 River Routes

24. A vertex-edge-vertex path is a sequence of zero or more edges. We define the edge_paths function – recursively.

25. That is, the empty sequence, $\langle\rangle$, is a vertex-edge-vertex path, [the first basis clause].

26. If e is an edge of g, and if (vi,vj) is in the mereology of e, then the $\langle(vi,ej,vk)\rangle$, where ej is the unique identifier of e is a vertex-edge-vertex path.

27. If p and p' are paths of g such that the last vertex identifier of the last element of p is the same as the first vertex identifier of the first element of p', then the sequence p followed by the sequence p' is a vertex-edge-vertex path of g [the inductive clause].

28. Only such paths which can be constructed by the above rules are edge paths [the extremal clause].

type
24. EP = E$^\omega$
24. edge_paths: G → EP-**set**
24. edge_paths(g) ≡
25.   **let** ps = {⟨⟩}
26.   ∪ {⟨(vi,uid_E(e),vk)⟩|e:E•e ∈ xtr_Es(g)∧(vi,vk)∈mereo_E(e)}
27.   ∪ {p^p′|p,p′:EP•{p,p′}⊆ps∧lVIlEP(9)=fVIfEP(p′)} **in**
24.   ps **end**

3.4 Rivers: Attributes

This author is not "an expert" on neither geograhical matters relating to rivers, lakes, etc., nor on the management of rivers: flood control, river taffic, etc. So, please, do not expect a very illuminating set of river attribute examples. All the attribute specifications are "tuned" to the purpose of the ensuing domain desription: whether for one or another form of river system study or eventual software system realisation.

29. River entities have geodetical positions –

30. all three dimensions: longitude, latitude and altitude[5].

31. River entities cover geodetical areas[6].

32. River entities have normal, low, high and overflow water levels[7].

33. River channels have "extent" in the form, for example of a precise description[8] of its course.[9]

[5] These are facts: How we represent them is a matter for geographers. Also: What is really mean by the 'position' of a source, or a river channel, etc. ? Also that is left for others to care about !

[6] See Footnote 5.

[7] See Footnote 5.

[8] See Footnote 5. In any **domain description**, yes, a precise description – whether "computable" [i.e., realizable] or not !

[9] – in a subsequent **requirements prescription** the domain description's "precise" form is replaced by, for example, a reasonably detailed [and computable] three dimensional *Bézier curve* specification [`en.wikipedia.org/wiki/B%C3%A9zier_curve`].

34. Lakes have a precise [three dimensional] description of their form, ...
35. Locks have ... et cetera
36. Waterfalls ...
37. Dams ...
38. Towns ...
39. Sources ...
40. Confluences ...
41. Mouths ...
42. Compositions of these have respective unions of these attributes.

type
29. GeoPos = Long $\times$ Lat $\times$ Alt
30. Long, Lat, Alt
32. Area
32. LoWL = ..., NoWL = ..., HiWL = ..., OfWL = ...
33. Course = ...
34. LakeForm = ...
36. ...; 37. ...; 38. ...;39. ...; 40. ...; 41. ...; 42. ...
value
29. attr_GeoPos: (SO|KO|MO|SE|CE) $\to$ GeoPos
30. attr_Long: GeoPos $\to$ ..., attr_lat: GeoPos $\to$ ...,attr_Alt: GeoPos $\to$ ...
32. attr_Area: (SO|KO|MO|SE|CE) $\to$ Area
32. attr_(LoWL|NoWL|HiWL|OfWL): (SO|KO|MO|SE|CE) $\to$ LoWL|NoWL|HiWL|OfWL
33. attr_CH: CH $\to$ Course
34. attr_LakeForm: LA $\to$ LakeForm
35. attr_...: LO $\to$ ...; 36. attr_...: WF $\to$ ...; 37. attr_...: DA $\to$ ...; 38. attr_...: TO $\to$ ...;
39. attr_...: SO $\to$ ...; 40. attr_...: KO $\to$ ...; 41. attr_...: MO $\to$ ...; 42. attr_...: ... $\to$ ...

We illustrate the issue of river attributes primarily to show you the sheer size and complexity of the task !

43. River entities have positions “within” their areas[10].
44. No two distinct river entities have conflicting (?) areas[11].
45. Two mereologically immediately adjacent river entities have bordering areas[12].
46.
47.

Axiom 44 is rather “sweeping”. It implies, of course, that river channels do not cross one another; that two or more non-channel river entities similarly do not “interfere” with one another, i.e., are truly “separate”.

[10] See Footnote 5.
[11] For example: their areas do not overlap. See Footnote 5.
[12] See Footnote 5.

4 Canals: The Endurants

As an example we wish our model to include the Dutch system of *polders*, *pumps*, *canals*, *locks*, *dikes*, *flood barriers*, *lakes*, *storm barriers* and the *ocean*.[13]

4.1 Some Introductory Remarks

4.1.1 The Dutch Polder System

We refer to Figs. 12 to 15 on Pages 15–17.

Figure 11: The Dutch Polder System

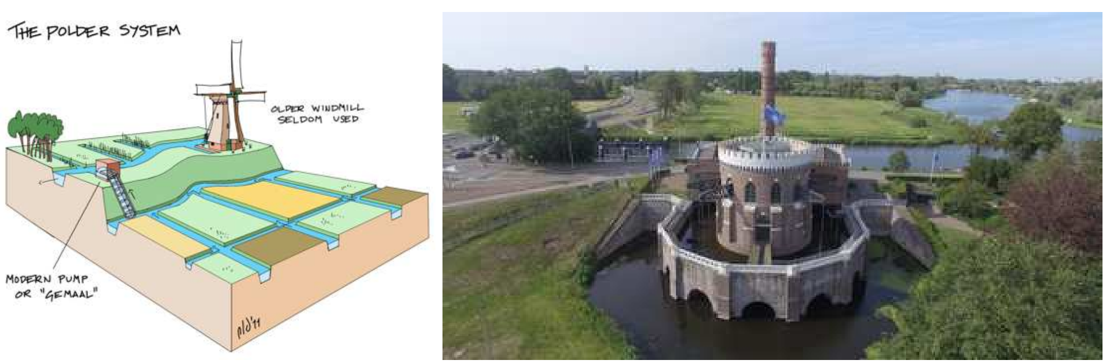

Figure 12: A Polder Schematic and The De Cruquius Pump

4.1.2 Natural versus Artefactual Domains

In contrast to river nets modeled earlier in this report, a system of mostly natural endurants, canal systems of *polders*, *pumps*, *canals*, *locks*, *dikes*, *flood barriers*, *lakes*, *storm barriers* and the *ocean* are, in a sense, dominated by man-made, i.e., artefactual endurants.

[13] www.fao.org/fileadmin/templates/giahs/PDF/Dutch-Polder-System_2010.pdf

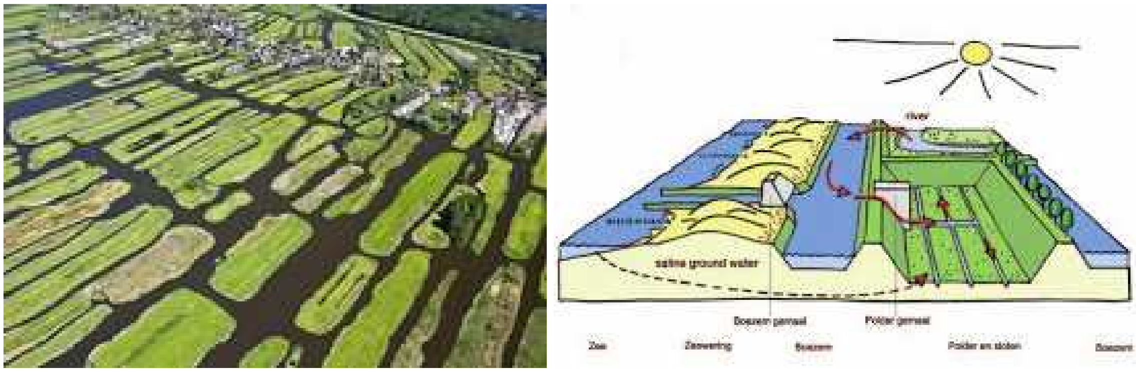

Figure 13: A Polder. Another Polder Schematic

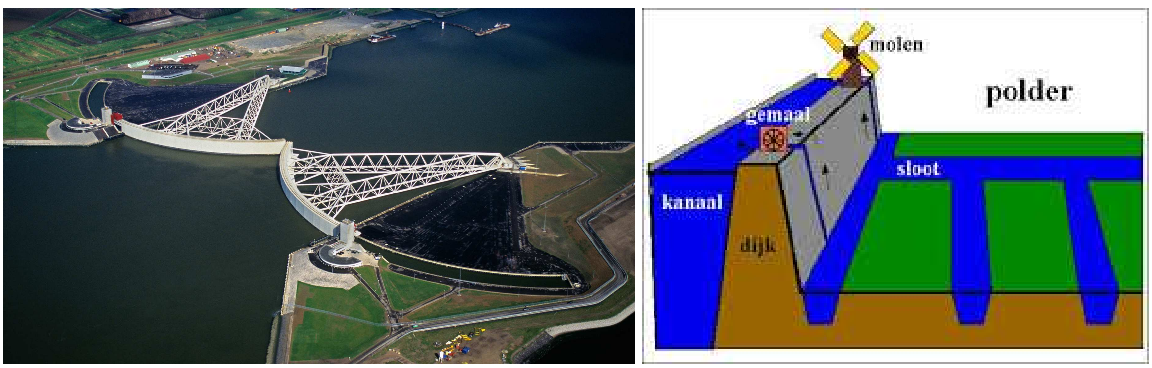

Figure 14: A Barrier. A Final Polder Schematic

4.1.3 Editorial Remarks

In order to develop an appropriate domain analysis & description of a reasonably comprehensive and representative canal domain I need answers to the following questions – and may more that can be derived from answer to these questions:

- **Canal Flow:** Are canals generally stagnant, or do canal water flow, that is, do canal flow have a preferred direction ?
- **Canal Graphs:** Do a canal form a[n undirected] graph, i.e., can canals be confluent with other canals ?
- **Canals and Rivers:** It is assumed that canals can be confluent with rivers. When canals join a river is it always with a lock of the canal onto the river – and the river flow is basically not interfered with by that canal, or otherwise ?
- **Canal Levels:** Can a canal pass a river overhead ? Or otherwise ? Same for canals and roads. Do canals run through mountains or over valleys ?

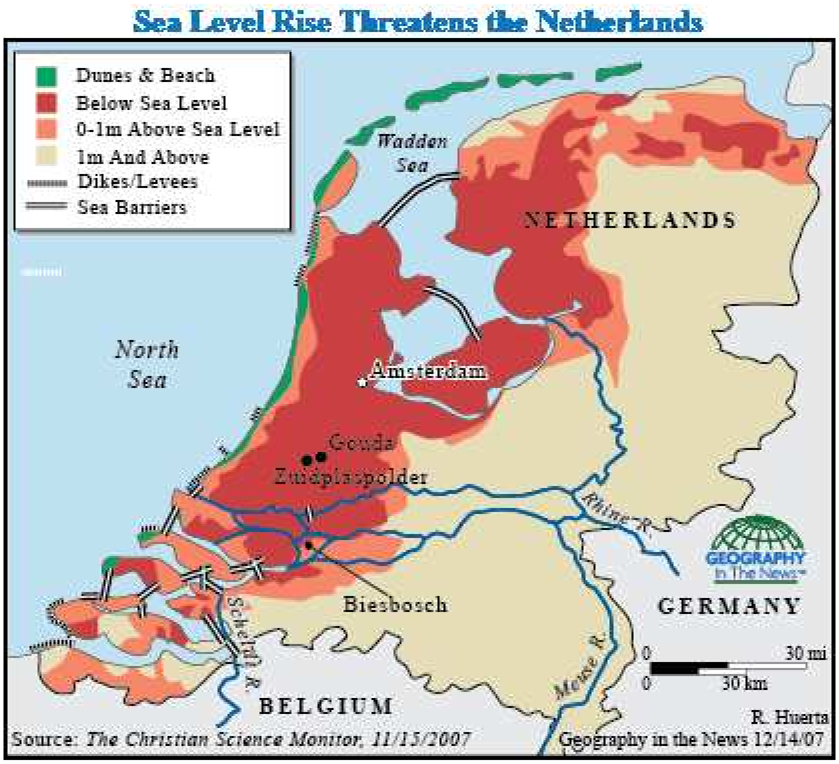

Figure 15: The Land-Water Levels of The Netherlands

- **Canal Locks:** It is assumed that an otherwise "unhindered" stretch of canal can have one or more locks. Yes or no ?
- **Canal Pumps:** It is assumed that there are two kinds of canal pumps: those in connection with locks, and those **not** in connection wit any locks. Yes or no ? I need information about the latter.
- **Polders and Pumps:** Are polders predominantly characterisable in terms of their land area and the pumps that keep these dry ?
- **Pumps and Canals:** Do polder pumps always operate in the context of canals ?

4.1.4 A Broad Sketch Narrative of Canal System Entities

- We take our departure point in the polders: So a polder-etc.-canal system contains **polders** and **polder pumps** take the water out of the polders and "puts" it in higher level **canals**.
- **Canals** are modeled as an undirected [general] graph whose vertices are **canal entities** and whose edges are given by the mereology of these entities – as to how they are topologically connected.
- The following are **canal entities:**

- **canal channels:** like river channels, only artefactual;
- **canal locks:** the locks as illustrated earlier;
- **canal pumps:** pumps water into locks – rather than using water from higher level canals;
- **canal gates:** protects the interior from ocean storm surges.

4.1.5 A Plan for The Canal System Description

Our plan is to analyse & describe

- external qualities of canal system endurants, Sect. 4.2.
- internal qualities of canal system, Sect. 4.3.1
- internal qualities of canal system, Sect. 4.3.2
- internal qualities of canal system, Sect. 4.3.4

For each of these categories we analyse & describe

- **sort**s and **type**s of these entities: endurants, unique identifiers, mereologies and attributes;
- **observer functions**, i.e., **obs_**$\cdots$, **uid_**$\cdots$ and **attr_**$\cdots$ for the observance of endurants, their unique identifiers, their mereologies and their attributes;
- **auxiliary functions** and
- **well-formedness predicates is_wf_**$\cdots$.

The external and internal quality definitions should be so conceived by the domain analyser & describer as to capture an essence, if not "the essence", of endurants. But they can never capture the essence "completely". As for the relation between *context free grammars* and *context sensitive grammars*, we must therefore introduce the notion of **well-formedness axioms**. The axioms constrain the relations between external and the various categories of internal qualities. More specifically:

48. The well-formedness of a canal system, is_wf_CS [48] is the conjunction of the well-formedness of canal system identifiers, is_wf_CS_Identities [109 on page 28], mereologies, is_wf_CS_Mereology [150a on page 35], and attributes, is_wf_CS_Attributes [277 on page 53].

type
48. CS
value
48. is_wf_CS: CS $\rightarrow$ **Bool**
48. is_wf_CS(cs) $\equiv$ is_wf_CS_Identifiers(cs)$\wedge$is_wf_CS_Mereologies(cs)$\wedge$ is_wf_CS_Attributes(cs)

4.1.6 No Structures

In this (long and detailed) example domain analysis & description I shall not use the pragmatic "device" of *structures* [cf, [1, *Sect.4.10*]]. Everything will be painstakingly analysed and described.

Some clarifying comments are in order:

- Compound endurants are either
  - *Cartesian*[14] or
  - *sets*.
- In analysing Cartesians, say c, into composite endurants, we analyse c into a number of components, $c_i, c_j, ..., c_k$, of respective sorts, $C_i, C_j, ..., C_k$, by means of observers obs_Ci, obs_Cj, ..., obs_Ck.
  - The Cartesians, C, in this report, all have:
    - unique identifiers,
    - mereologies and
    - attributes.
  - So do each of the $C_i, C_j, ..., C_k$.
- In analysing an endurant, E, into sets, say s or sort S, we first analyses E into a separately observable endurant Ss, i.e., obs_Ss, which we then, at the same time define as Ss = S-**set**.
- An Ss endurant thus has all the internal qualities:
  - a unique identifier,
  - a mereology
  - and attributes.

4.1.7 Sequences of Presentation

The sequence in which endurant sorts are introduced is "repeated" in the sequences in which unique identifier sorts and mereology types are introduced. Thus the sequences of narrative and formal

- endurant items,
  - sort items, Items 49–63 on page 22,
  - value items, Items 69–88 on Pages 23–25 and
  - "alternative" value items, Items ν69–ν88 on Pages 25–26,

are "repeated" in

- unique identifier
  - sort items, Items 89–107 on Pages 26–27,
  - value items, Items 109–124 on Pages 28–29 and

[14] Cartesian is spelled with a large 'C', after René Descartes, the French mathematician (1596–1650) `https://da.wikipedia.org/wiki/René_Descartes`.

  - "alternative" value items, Items $v63$–$v76$ on Pages 29–30,

and in

- mereology
  - type items, Items 132 to 148 on Pages 30–35 and
  - well-formedness items, Items 150 to 168 on Pages 35–42.

4.1.8 Naming Conventions

Some care has been taken in order to name endurants, including sets of and predicates and functions over these; their unique identifiers and typed sets and values of and predicates and functions over these; their mereologies and typed sets and values of and predicates and functions over these; and their attributes and typed sets and values of and predicates and functions over these.

4.2 Canals: External Qualities

4.2.1 Endurant Sorts

The narrative(s) that follow serves two purposes:

- a formal purpose: the identification of endurants, and
- an informal purpose: in "casually familiarising" the reader as the the rôle of these endurants.

The former purpose is the only one to formalise. The latter purpose informally "herald" things to come – motivating, in a sense, theses "things", the internal qualities and, if we had included a treatment of canal perdurants, the behaviours of these canal elements seen as behaviours.

All the elements mentioned below consist of both discrete endurants and fluids, i.e., water. In contrast to the treatment of such conjoins in [1, Sect. 4.13.3] we shall, in an informal digression from the principles, techniques and tool of the *analysis & description calculi* of [1, Chapters 4–5], omit "half the story" ! It will be partly "restored" in out treatment of *canal attributes*, Sect. 4.3.4.

In this section we shall narrate all the different endurant sorts, Items 49–68 (Pages 20–22), before we formalise them (Pages 22–22). We beg the readers forebearance in possibly having to thumb between narrative (page)s and formalisations (page)s.

49. **Canal systems**, CS, are given.
50. From a canal system one can observe a **canal net**, CN.
51. From a canal system one can observe a **polder aggregate**, PA.

Observing two endurants of a composite endurant is as if the composite is a Cartesian product of two. Hence the "(...,...)" of Fig. 16 on page 23.

52. From canal nets one can observe **canal hub aggregates**, CA_HA, and
53. **canal link aggregates**, CA_LA.
54. From a polder aggregate one can observe a **polder set**, Ps, of polders, P. One observes the set, not its elements.

55. From a canal hub aggregate one can observe a **hub set**, CA_Hs, of hubs, CA_H. One observes the set, not its elements.

56. From a canal link aggregate one can observe a **canal link set**, CA_Ls, of canal links, CA_L. One observes the set, not its elements.

57. Polders are considered atomic. A **polder** is a low-lying tract of land that forms an artificial hydrological entity, enclosed by embankments known as dikes. The three types of polder[15] are:
    - Land reclaimed from a body of water, such as a lake or the seabed.
    - Flood plains separated from the sea or river by a dike.
    - Marshes separated from the surrounding water by a dike and subsequently drained; these are also known as koogs.

58. Canal hubs are considered atomic and are:

59. either **canal begin/end**s (that is, where there is no continuation of a canal: where it ends "blind', or where begins "suddenly'[16]), CA_BE,

60. or **canal confluence**s (of three or more canals[17]), CA_CO,

61. or **canal outlet**s, CA_OU (where canals join a *river*, or a *lake*, or an *ocean*). These sorts are all considered atomic.

62. **Canal links** are aggregates.

63. From canal links we choose to observe a set of **canal link element**s, CA_LE[18]. (Canal links are such, through their mereology, see Sect. 4.3.2, that they form two reversible sequences between connecting edges.)

64. Canal link elements are considered atomic and are

65. either **canal channel**s, CA_CH[19],

66. or **canal lock**s, CA_LO[20],

---

[15] The ground level in drained marshes subsides over time. All polders will eventually be below the surrounding water level some or all of the time. Water enters the low-lying polder through infiltration and water pressure of groundwater, or rainfall, or transport of water by rivers and canals. This usually means that the polder has an excess of water, which is pumped out or drained by opening sluices at low tide. Care must be taken not to set the internal water level too low. Polder land made up of peat (former marshland) will sink in relation to its previous level, because of peat decomposing when exposed to oxygen from the air.

Polders are at risk from flooding at all times, and care must be taken to protect the surrounding dikes. Dikes are typically built with locally available materials, and each material has its own risks: sand is prone to collapse owing to saturation by water; dry peat is lighter than water and potentially unable to retain water in very dry seasons. Some animals dig tunnels in the barrier, allowing water to infiltrate the structure; the muskrat is known for this activity and hunted in certain European countries because of it. Polders are most commonly, though not exclusively, found in river deltas, former fenlands, and coastal areas.

[16] A canal "end" is a canal channel which is "connected" only at one end to a canal channel.

[17] Without loss of generality we model only confluences of three canals.

[18] We could have chosen other abstractions, for example, to observe a sequence of elements. More on this later.

[19] A canal channel offers a "straight", un-interrupted "stretch" of water – like does a river channel.

[20] A canal lock) is always connected to two distinct canal link elements. Canal locks still act like a waterway, as does a canal channel.

67. or **canal polder pump**s, CA_PO_PU[21].

68. We do not further describe **canal outlets, rivers, lakes** and **oceans**.

```
type
49.  CS
50.  CN
51.  PA
52.  CA_HA
53.  CA_LA
54.  Ps = P-set, P
55.  CA_Hs = CA_H-set
56.  CA_Ls = CA_L-set
57.  P
58.  CA_H == CA_BE|CA_CO|CA_OU
59. CA_BE :: ...
60. CA_CO :: ..
61. CA_OU :: ..
62. CA_L
63.  CA_LEs = CA_LE-set
64.  CA_LE == CA_CH|CA_LO|CA_LO_PU|CA_PO_PU
65. CA_CH :: ...
66. CA_LO :: ...
67. CA_PO_PU :: ...
value
50.  obs_CN: CS → CN
51.  obs_PA: CS → PA
52.  obs_CA_HA: CS → CA_HA
53.  obs_CA_LA: CN → CA_LA
54.  obs_Ps: PA → Ps
55.  obs_CA_Hs: CA_HA → CA_Hs
56.  obs_CA_Ls: CA_LA → CA_Ls
63.  obs_CA_LEs: CA_L → CA_LEs
```

Figure 16 on the facing page shows the ontology of a wide class of canal systems.

Figure 17 on page 24 shows the schematisation of a specific canal system.

Figure 18 on page 25 shows the individual endurants of a canal system for that shown in Fig. 17 on page 24. Given what we have formalised so far, i.e., formula 49–56, this is really all we can "diagram". The "part" list of Fig. 18 on page 25 cannot show other than that there are these parts, but not how they are connected – that is first revealed when we ascribe mereologies – and that there are canal channels, not, for example, their length – that is first revealed when we ascribe attributes, such as length.

[21] A canal polder pump is a pump that takes water from a polder and deposits it in a canal which is at a higher level than the polder.

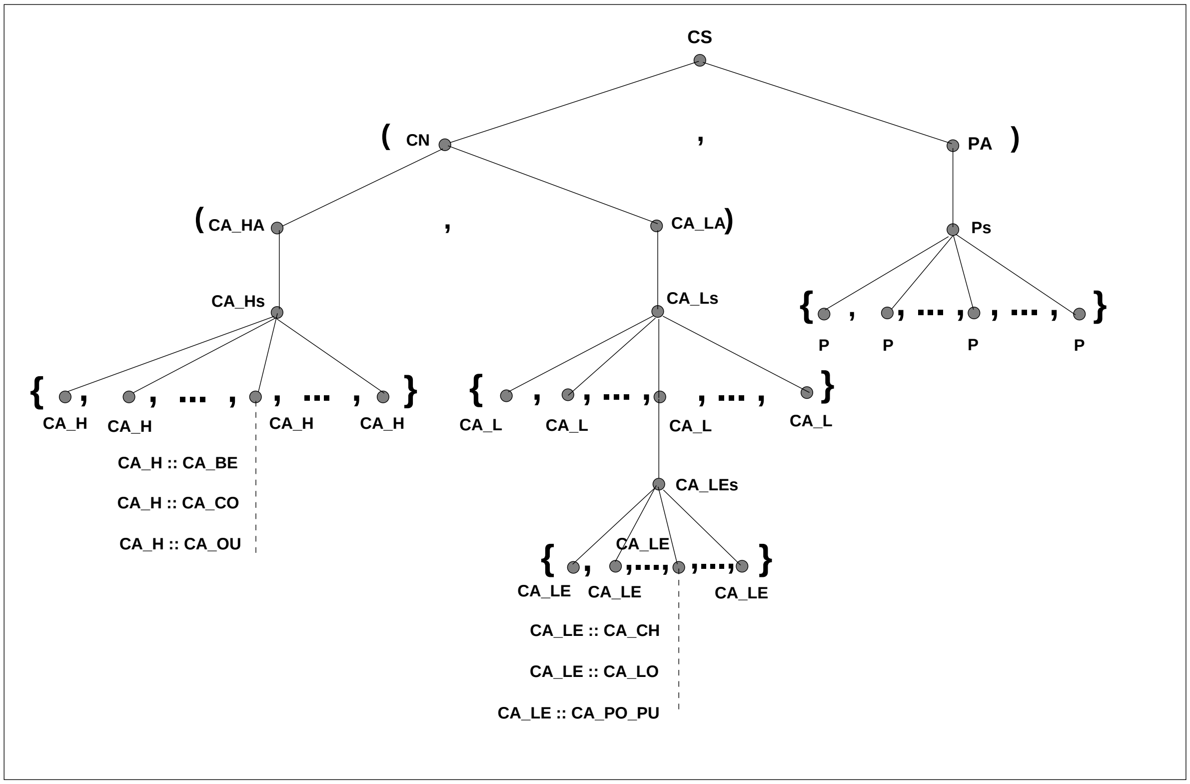

Figure 16: Canal System Ontology

4.2.2 Some Calculations

We refer to Fig. 19 on page 26. We shall list the endurant parts – and later on their unique identifierss in the left-to-right order of a breadth-first traversal of the canal ontology.

69. Let cs be a “global” canal system.[22]
70. Canal nets and polders can be seen as consisting of the following endurants, modeled as a map, map_ends:
71. the canal system, cs_{end}, [ι 49 π 20]
72. the canal net, cn_{end}, [ι 50 π 20]
73. the polder aggregate, pa_{end}, [ι 51 π 20]
74. the canal net hub aggregate, ca_ha_{end}, [ι 52 π 20],
75. the canal net link aggregate, ca_la_{end}, [ι 53 π 20]

[22] Introducing cs allows us to refer to it and its “derivatives”, “all over”, and thus “universally prefix quantify” many axioms.

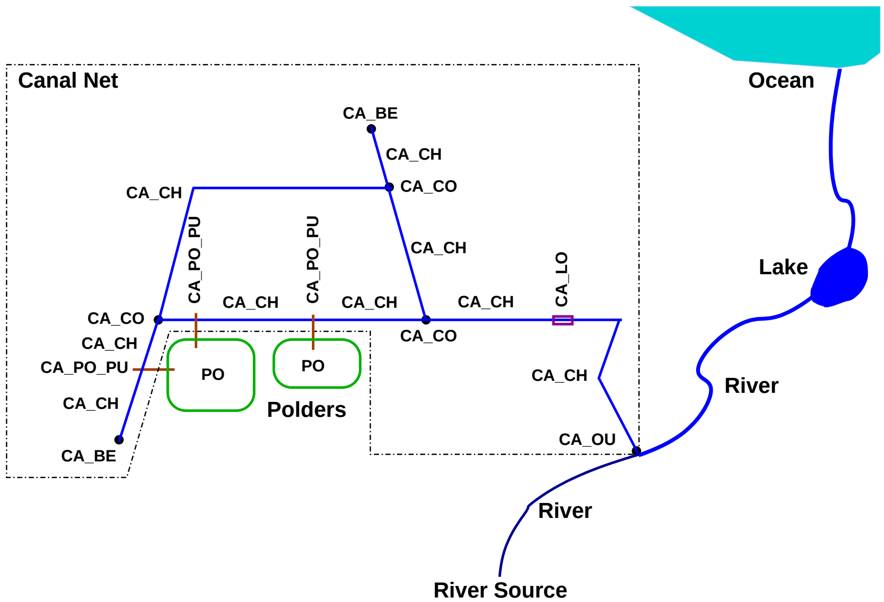

Figure 17: A Schematised Specific Canal System: Canal Net + Polders

76. the set of polders, ps_{end}, [ι 54 π 20]
77. the set of hubs, ca_hs_{end}, [ι 55 π 21]
78. the set of canal links, ca_ls_{end}, [ι 56 π 21]
79. the set of polders, pos_{end}, [ι 57 π 21]
80. the set of hubs, ca_hs_{end}, [ι 55 π 21] have the following kinds of hubs: [ι 58 π 21]
81. canal begin/ends, ca_bes_{end}, [ι 59 π 21],
82. canal confluences, ca_cos_{end}, [ι 60 π 21] and
83. canal outlets, ca_ous_{end}, [ι 61 π 21],
84. the set of canal link elements, ca_les_{end} [ι 63 π 21],
85. the canal link elements [ι 64 π 21] are of the following kinds:
86. canal channels, ca_chs_{end}, [ι 65 π 21]
87. canal locks, ca_los_{end}, [ι 66 π 21]
88. canal polder pumps, $ca\_po\_pus_{end}$ [ι 67 π 22].

value
69. *cs*:CS
70. map_ends: MAP_END[23]

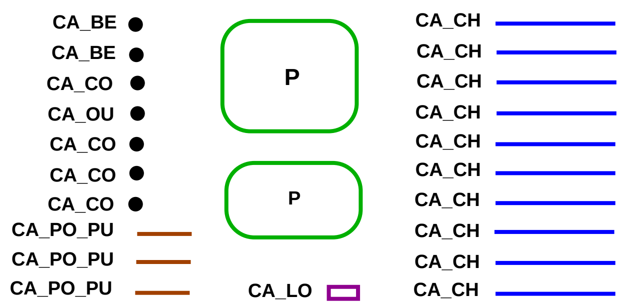

Figure 18: Component Endurants of the Canal System of Fig. 17

70. map_ends = [
71. $cs_{end} \mapsto \{cs\}$, [ι 49 π 20]
72. $cn_{end} \mapsto \{\text{obs\_CN}(\text{map\_ends}(cs_{end}))\}$, [$\iota$ 50 π 20]
73. $pa_{end} \mapsto \{\text{obs\_PA}(\text{map\_ends}(cs))\}$, [$\iota$ 51 π 20]
74. $ca\_ha_{end} \mapsto \{\text{obs\_CA\_HA}(\text{map\_ends}(cs))\}$, [$\iota$ 52 π 20]
75. $ca\_la_{end} \mapsto \{\text{obs\_CA\_LA}(\text{map\_ends}(cn_{end}))\}$, [$\iota$ 53 π 20]
76. $ps_{end} \mapsto \text{obs\_Ps}(\text{map\_ends}(pa_{end})$, [$\iota$ 54 π 20]
77. $ca\_hs_{end} \mapsto \text{obs\_CA\_Hs}(\text{map\_ends}(ca\_pa_{end}))$, [$\iota$ 55 π 21]
78. $ca\_ls_{end} \mapsto \text{obs\_CA\_Ls}(\text{map\_ends}(ca\_ps_{end}))$, [$\iota$ 56 π 21]
79. $pos_{end} \mapsto \text{map\_ends}(ca\_ps_{end}))$, [$\iota$ 57 π 21]
81. $ca\_bes_{end} \mapsto \text{map\_ends}(ca\_hs_{end}) \setminus \text{CA\_BE}$, [$\iota$ 59 π 21]
82. $ca\_cos_{end} \mapsto \text{map\_ends}(ca\_hs_{end}) \setminus \text{CA\_CO}$, [$\iota$ 60 π 21]
83. $ca\_ous_{end} \mapsto \text{map\_ends}(ca\_hs_{end}) \setminus \text{CA\_OU}$, [$\iota$ 61 π 21]
84. $ca\_les_{end} \mapsto \text{obs\_CA\_Ls}(\text{map\_ends}(ca\_ps_{end}))$, [$\iota$ 56 π 21]
86. $ca\_chs_{end} \mapsto \cup\, \text{map\_ends}(ca\_les_{end}) \setminus \text{CA\_CH}$, [$\iota$ 65 π 21]
87. $ca\_los_{end} \mapsto \cup\, \text{map\_ends}(ca\_les_{end}) \setminus \text{CA\_LO}$, [$\iota$ 66 π 21]
88. $ca\_po\_pus_{end} \mapsto \cup map\_ends(ca\_les\_ps_{end}) \setminus \text{CA\_PO\_PU}$][ι 67 π 22]

We, in a name-overloading fashion, define – note the v prefix of the formula item numbers:

value
v71. $cs_{end} = cs$, [ι 49 π 20]
v72. $cn_{end} = \text{obs\_CN}(\text{map\_ends}(cs_{end}))$, [$\iota$ 50 π 20]
v73. $pa_{end} = \text{obs\_PA}(\text{map\_ends}(cs))$, [$\iota$ 51 π 20]
v74. $ca\_ha_{end} = \text{obs\_CA\_HA}(\text{map\_ends}(cs))$, [$\iota$ 52 π 20]
v75. $ca\_la_{end} = \text{obs\_CA\_LA}(\text{map\_ends}(cn_{end}))$, [$\iota$ 53 π 20]
v76. $ps_{end} = \text{obs\_Ps}(\text{map\_ends}(pa_{end}))$, [$\iota$ 54 π 20]
v77. $ca\_hs_{end} = \text{obs\_CA\_Hs}(\text{map\_ends}(ca\_pa_{end}))$, [$\iota$ 55 π 21]
v78. $ca\_ls_{end} = \text{obs\_CA\_Ls}(\text{map\_ends}(ca\_la_{end}))$, [$\iota$ 56 π 21]

[23] We invite the reader to formulate the MAP_END type. As you can see from Items 60–70, it is a map from some sort of names to sets of endurants.

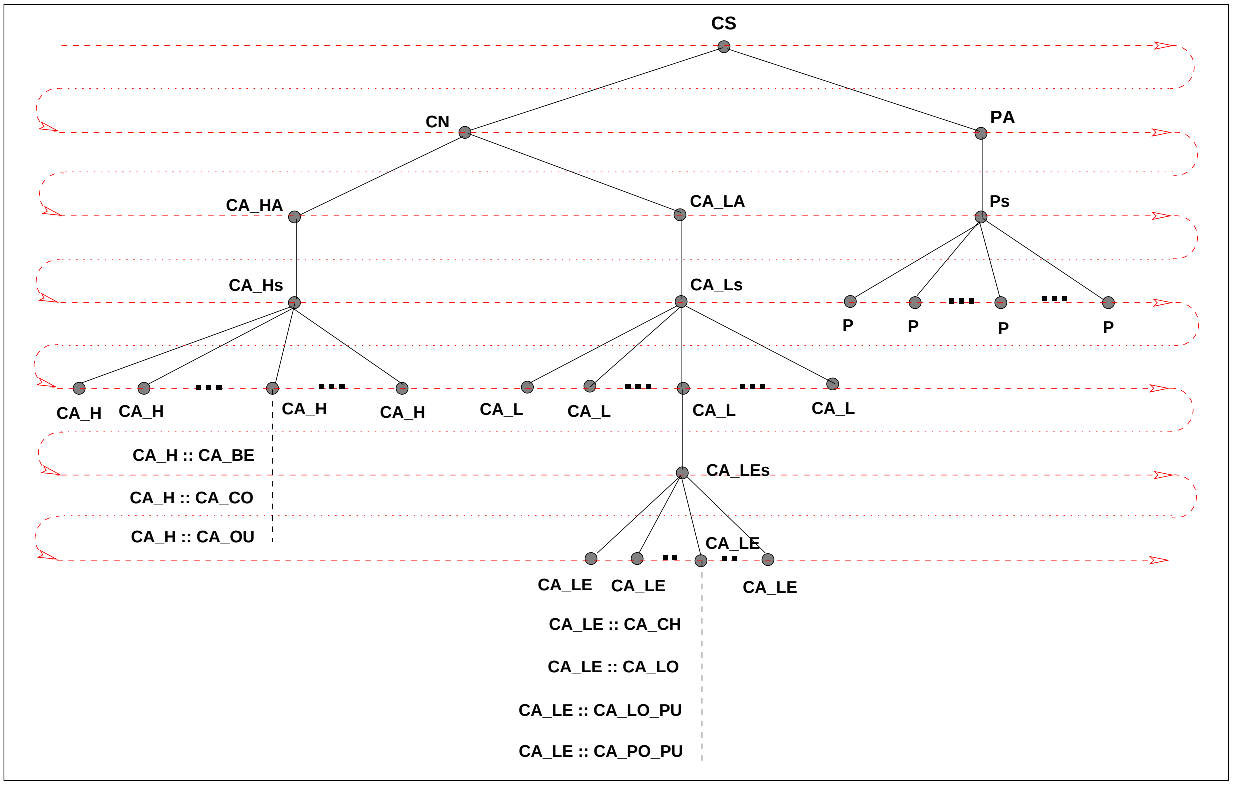

Figure 19: A Breadth-first Left-to-Right [Top-down] Canal Ontology Traversal

v79. $pos_{end} = \text{map\_ends}(ca\_ps_{end})$, [ι 57 π 21]
v81. $ca\_bes_{end} = \text{map\_ends}(ca\_hs_{end}) \setminus \text{CA\_BE}$, [ι 59 π 21]
v82. $ca\_cos_{end} = \text{map\_ends}(ca\_hs_{end}) \setminus \text{CA\_CO}$, [ι 60 π 21]
v83. $ca\_ous_{end} = \text{map\_ends}(ca\_hs_{end}) \setminus \text{CA\_OU}$, [ι 61 π 21]
v84. $ca\_cles_{end} = \text{obs\_CA\_LEs}(\text{map\_ends}(ca\_ls_{end}))$, [ι 63 π 21]
v86. $ca\_chs_{end} = \cup\, \text{map\_ends}(ca\_les_{end}) \setminus \text{CA\_CH}$, [ι 65 π 21]
v87. $ca\_los_{end} = \cup\, \text{map\_ends}(ca\_les_{end}) \setminus \text{CA\_LO}$, [ι 66 π 21]
v88. $ca\_po\_pus_{end} = \cup map\_ends(ca\_les\_ps_{end}) \setminus \text{CA\_PO\_PU}$, [ι 67 π 22]

4.3 Canals: Internal Qualities

4.3.1 Canals: Unique Identifiers

4.3.1.1 Unique Identifier Sorts:

89. Canal systems have unique identifiers [ι 71 π 23].
90. Canal nets have unique identifiers [ι 72 π 23].
91. Polder aggregates have unique identifiers [ι 73 π 23].
92. Canal hub aggregates have unique identifiers [ι 74 π 23].
93. Canal link aggregates have unique identifiers [ι 75 π 23].
94. Polder sets (of polders) have unique identifiers [ι 76 π 24].
95. Canal hub sets have unique identifiers [ι 77 π 24].
96. Canal link sets have unique identifiers [ι 78 π 24].

97. Polders have unique identifiers.
98. Canal hubs have unique identifiers:
99. canal begin/ends [ι 81 π 24],
100. canal confluences [ι 82 π 24] and
101. canal outlets [ι 83 π 24].
102. Canal links have unique identifiers [ι 62 π 21].
103. Canal link element sets have unique identifiers [ι 84 π 24].
104. Canal link elements have unique identifiers:
105. canal channels [ι 86 π 24],
106. canal locks [ι 87 π 24] and
107. canal polder pumps [ι 88 π 24].

type
89. CS_UI [ι 49 π 20]
90. CN_UI [ι 50 π 20]
91. PA_UI [ι 51 π 20]
92. CA_HA_UI [ι 52 π 20]
93. CA_LA_UI [ι 53 π 20]
94. Ps_UI [ι 54 π 20]
95. CA_Hs_UI [ι 55 π 21]
96. CA_Ls_UI [ι 56 π 21]
97. P_UI [ι 57 π 21]
98. CA_H_UI = [ι 58 π 21]
98. CA_BE_UI|CA_CO_UI|CA_OU
99. CA_BE_UI [ι 59 π 21]
100. CA_CO_UI [ι 60 π 21]
101. CA_OU_UI [ι 61 π 21]
102. CA_L_UI [ι 62 π 21]
103. CA_LEs_UI [ι 63 π 21]
104. CA_LE_UI = CA_CH_UI [ι 64 π 21]
104. |CA_LO_UI|CA_LO_PU_UI|CA_PO_PU_UI
105. CA_CH_UI [ι 65 π 21]
106. CA_LO_UI [ι 66 π 21]
107. CA_PO_PU_UI [ι 67 π 22]

value
89. uid_CS: CS→ CS_UI
90. uid_CN: CN − > CN_UI
91. uid_PA: PA → PA_UI
92. uid_CA_HA: CA_HA → CA_HA_UI
93. uid_CA_LA: CA_LA → CA_LA_UI
94. uid_Ps: Ps → Ps_UI
95. uid_CA_Hs: CA_Hs → CA_Hs_UI
96. uid_CA_Ls: CA_Ls → CA_Ls_UI
97. uid_P: P → P_UI
99. uid_CA_BE: CA_BE → CA_BE_UI
100. uid_CA_CO: CA_CO → CA_CO_UI
101. uid_CA_OU: CA_OU → CA_OU_UI
102. uid_CA_L: CA_L → CA_L_UI
103. uid_CA_LEs: CA_LEs → CA_LEs_UI
105. uid_CA_CH: CA_CA_CH → CA_CH_UI
106. uid_CA_LO: CA_CA_LO → CA_CA_LO_UI
107. uid_CA_PO_PU: CA_LE → CA_PO_PU_UI

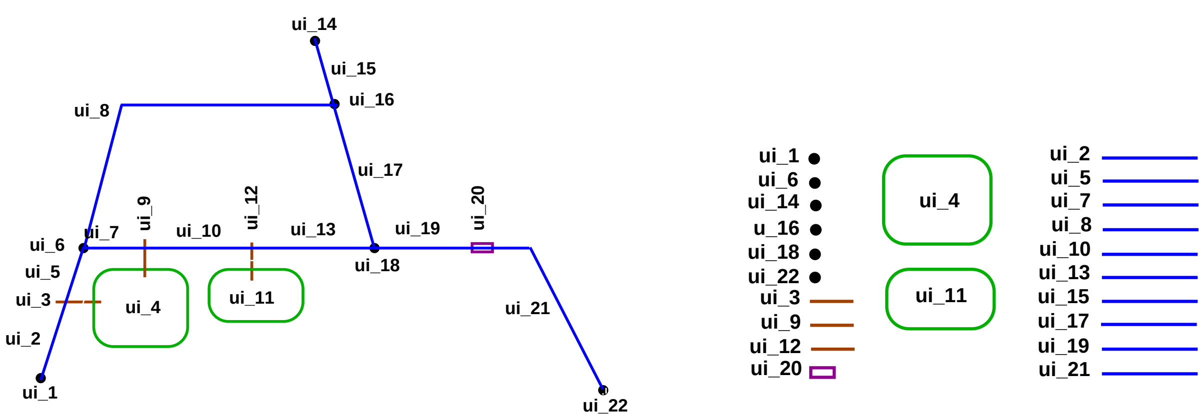

Figure 20: Unique Identifiers of the Canal System of Figs. 17 and 18

4.3.1.2 Some Calculations:

108. We can calculate the following sets of unique identifiers, seen as a map from some kind of RSL names to sets of unique identifiers:

109. the canal system singleton set of its unique identifier, [ι 71 π 23],

110. the canal net singleton set of its unique identifier, [ι 72 π 23],

111. the polder aggregate singleton set of its unique identifier, [ι 73 π 23],

112. the canal hub aggregate singleton set of its unique identifier, [ι 74 π 23],

113. the canal link aggregate singleton set of its unique identifier, [ι 75 π 23],

114. the polder set singleton set of its unique identifier, [ι 76 π 24],

115. the hub set singleton set of its unique identifier, [ι 77 π 24],

116. the link set singleton set of its unique identifier, [ι 78 π 24],

117. the set of polder unique identifiers, [ι 76 π 24],

118. the set of canal begin/end unique identifiers, [ι 60 π 21],

119. the set of canal confluence unique identifiers, [ι 60 π 21],

120. the set of canal outlet unique identifiers, [ι 60 π 21],

121. the set of canal link set unique identifiers, [ι 61 π 21],

122. the set of canal link unique identifiers, [ι 62 π 21],

123. the set of canal channel unique identifiers, [ι 65 π 21],

124. the set of canal lock unique identifiers, [ι 66 π 21],

125. the set of canal polder pump unique identifiers, [ι 67 π 22].

To define the next map we make use of the following generic function:

126. It applies to a set of endurants of sort X and yields the set of unique identifiers of the members of that set.

type
126. uid_X: X-**set** $\rightarrow$ X_UI-**set**
value
126. uid_X(xs) $\equiv$ {uid_X(x)|x:X•x $\in$ xs}

value
108. map_uids: MAP_UI[24]
108. map_uids = [[ι 49 π 20]
109. cs_{uid} $\mapsto$ uid_CS(map_end(cs_{end})), [ι 71 π 23]
110. cn_{uid} $\mapsto$ uid_CN(map_ends(cn_{end})), [ι 72 π 23]
111. pa_{uid} $\mapsto$ uid_PA(map_ends(pa_{end})), [ι 73 π 23]

112. ca_ha_{uid} $\mapsto$ uid_CA_HA(map_ends(ca_ha_{end})) , [ι 74 π 23]
113. ca_la_{uid} $\mapsto$ uid_CA_LA(map_ends(ca_la_{end})) , [ι 75 π 23]
114. ps_{uid} $\mapsto$ uid_P(map_ends(ps_{end})), [ι 67 π 22]
115. ca_hs_{uid} $\mapsto$ uid_CA_Hs(map_ends(ca_hs_{end})) , [ι 77 π 24]
116. ca_ls_{uid} $\mapsto$ uid_CA_Ls(map_ends(ca_ls_{end})) , [ι 78 π 24]
117. pos_{uid} $\mapsto$ uid_P(map_ends(ps_{end})) , [ι 78 π 24]
118. ca_bes_{uid} $\mapsto$ uid_BE(map_ends(ca_bes_{end})), [ι 81 π 24]
119. ca_cos_{uid} $\mapsto$ uid_CO(map_ends(ca_cos_{end})), [ι 82 π 24]
120. ca_ous_{uid} $\mapsto$ uid_OU(map_ends(ca_ous_{end})), [ι 83 π 24]
121. ca_les_{uid} $\mapsto$ uid_CA_LEs(map_ends(ca_ls_{end})), [ι 84 π 24]
123. ca_chs_{uid} $\mapsto$ uid_CA_CH(map_ends(ca_chs_{end})), [ι 86 π 24]
124. ca_los_{uid} $\mapsto$ uid_CA_LO(map_ends(ca_los_{end})), [ι 87 π 24]
125. $ca\_po\_pus_{uid}$ $\mapsto$ uid_PO_PU(map_ends($ca\_po\_pus_{end}$))] [ι 88 π 24]

4.3.1.3 An Axiom:

127. Let end_{parts} stand for the set of all composite and atomic canal system endurants,

128. and end_{uids} the set of all their unique identifiers.

129. The number of endurants parts equals the number of endurant part unique idenifiers, is_wf_CS_Identities(cs).

value
127. $end_{parts} = \cup$ **rng** proper_map_ends
128. $end_{uids} = \cup$ **rng** map_uids
axiom
129. is_wf_CS_Identities: CS $\rightarrow$ **Bool**
129. is_wf_CS_Identities(cs) $\equiv$ **card** end_{parts} = **card** end_{uids}

4.3.1.4 Another Representation of UI Values:
We, in a somewhat name-overloading fashion, similarly define:

value
v109. cs_{uid} = uid_CS(cs_{end}), [ι 71 π 25]
v110. cn_{uid} = uid_CN(cn_{end}), [ι 72 π 25]
v111. pa_{uid} = uid_PA(pa_{end}), [ι 73 π 25]
v112. ca_ha_{uid} = uid_CA_HA(ca_ha_{end}) , [ι 74 π 25]
v113. ca_la_{uid} = uid_CA_LA(ca_la_{end}) , [ι 75 π 25]
v114. ps_{uid} = uid_Ps(ps_{end}), [ι 76 π 25]
v115. ca_hs_{uid} = uid_Hs(ca_pa_{end}), [ι 77 π 25]
v116. ca_ls_{uid} = uid_Ls(ca_ps_{end}), [ι 78 π 25]
v117. pos_{uid} = uid_P(ps_{end}), [ι 79 π 26]
v118. ca_bes_{uid} = uid_BE(ca_bes_{end}), [ι 81 π 26]
v119. ca_cos_{uid} = uid_CO(ca_cos_{end}), [ι 82 π 26]

[24] We invite the reader to formulate the MAP_UI type. As you can see from Items 60–70, it is a map from some sort of names to sets of unique identifiers.

ν120. ca_ous_{uid} = uid_OU(ca_ous_{end}), [ι 83 π 26]
ν121. ca_cles_{uid} = uid_CA_LEs(ca_les_{end}), [ι 84 π 26]
ν122. ca_chs_{uid} = uid_CA_CH(ca_chs_{end}), [ι 86 π 26]
ν123. ca_los_{uid} = uid_CA_LO(ca_los_{end}), [ι 87 π 26]
ν??. $ca\_po\_pus_{uid}$ = uid_PO_PU($ca\_po\_pus_{end}$)[ι 88 π 26]

4.3.1.5 An Extract Function:

130. Given 127. end_{parts} and 128. end_{uids}, we can, from any known unique identifier obtain its corresponding part:

value
130. get_part: UI → END
130. get_part(ui) ≡ **let** p:P • p ∈ end_{parts} • uid_P(p)=ui **in** p **end**; **pre**: ui ∈ **rng** end_{uids}

4.3.2 Canals: Mereologies

4.3.2.1 Mereology Types:

We shall focus only on the **topological mereologies** of canal system endurants. These can be "read off" the ontology tree of Fig. 16 on page 23. Had we included the modeling of vessels that ply the waters of canals, then the mereologies of most canal endurants wouldalso include sets of vessel identifiers.

As for the definitions of endurants, cf. Items 49 on page 20 to 67 on page 22, and the unique identifiers, cf. Items 89 on page 26 to 107 on page 27, we define the mereologies for each category of endurants. These mereologies are defined using the unique identifiers of the endurants immediately "above" and "below" them in the ontology "tree" of Fig. 16 on page 23.

4.3.2.1.1 Common Hub and Link Types:

From the unique identifier section we take over types defined in Items 91 and 92 on page 26

131. while introducing a set of their identifiers:

type
91. CA_HE_UI = CA_BE_UI|CA_CO_UI|CA_OU_UI
92. CA_LE_UI = CA_CH_UI|CA_LO_UI|LO_PU_UI|PO_PU_UI
131. CA_LE_UI_H = (CL_HE_UI|CP_LE_UI)-**set**

4.3.2.1.2 Canal Systems:

132. The mereology of a *canal system* is a pair of the unique identifiers of the canal net and of the polder aggregate.

type
132. CS_Mer = CN_UI × PA_UI
value
132. mereo_CS: CS → mereo_CS

4.3.2.1.3 Canal Nets:

133. The mereology of a *canal net* aggregate is a pair of the unique identifier of the canal system, of which it is a part, and a pair of the set of the unique identifiers of the canal hub agregate and the canal link aggregate of the net.

type
value
133. $\text{CN\_Mer} = \text{CS\_UI} \times (\text{CA\_HA} \times \text{CA\_LA})$
value
133. $\text{mereo\_CN: CN} \rightarrow \text{CN\_Mer}$

4.3.2.1.4 Polder Aggregates:

134. The mereology of a *polder* aggregate is a pair of the unique identifier of the canal system, of which it is a part, and the unique identifier of the polder set it "spawns".

type
134. $\text{PA\_Mer} = \text{CS\_UI} \times \text{Ps\_UI}$
value
134. $\text{mereo\_PA: PA} \rightarrow = \text{PA\_Mer}$

4.3.2.1.5 Canal Hub Aggregates:

135. The mereology of a *hub aggregate* is a pair of the unique identifier of the canal net it belongs to and the hub set it "spawns".

type
135. $\text{CA\_HA\_Mer} = \text{CN\_UI} \times \text{CA\_Hs}$
value
135. $\text{mereo\_CA\_HA: HA} \rightarrow \text{CA\_HA\_Mer}$

4.3.2.1.6 Canal Link Aggregates:

136. The mereology of a *link aggregate* is a pair of the unique identifier of the canal net it belongs to and a set of the unique identifiers of the links that it "spawns".

type
136. $\text{CA\_LA\_Mer} = \text{CN\_UI} \times \text{CA\_Ls}$
value
136. $\text{mereo\_CA\_LA: LA} \rightarrow \text{CA\_LA\_Mer}$

4.3.2.1.7 Sets of Polders:

137. The mereology of a *polder set* is a pair of the unique identifier of the polder aggregate it belongs to and a set of the unique identifiers of the polders that it “spawns”.

type
137. Ps_Mer = PA_UI × P_UI-**set**
value
137. mereo_Ps: Ps → Pa_Mer

4.3.2.1.8 Sets of Hubs:

138. The mereology of a *hub set* is a pair of the unique identifier of the hub aggregate it belongs to and a set of the unique identifiers of the hubs that it “spawns”.

type
138. CA_Hs_Mer = CA_HA_UI × CA_H_UI-**set**
value
138. mereo_CA_Hs: CA_Hs → CA_Hs_Mer

4.3.2.1.9 Sets of Links:

139. The mereology of a *link set* is a pair of the unique identifier of the link aggregate it belongs to and a set of the unique identifiers of the links that it “spawns”.

type
139. CA_Ls_Mer = CS_LA_UI × CA_L_UI-**set**
value
139. mereo_CA_Ls: Ls → CA_Ls_Mer

4.3.2.1.10 Polders:

140. The mereology of a*polder* is a pair of the unique identifier of the polder aggregate and a set of unique identifiers of *canal polder pump*s.

type
140. P_Mer = Ps_UI
value
140. mereo_P: P → P_Mer

4.3.2.1.11 Hubs:

- Hubs are not individually "recognisable" as such. They are either begin/ends, confluences or outlets; cf. Item 58 on page 21.
- The mereologies of hubs thus "translates" into the mereology of either begin/ends, confluences or outlets.

4.3.2.1.11.1 Begin/End:

141. The mereology of a *canal begin/end* is a pair: the unique identifier of the canal hub set it belongs to and the singleton set of the unique identifier of the first canal link element for which it is the begin/end.

type
141. CA_BE_Mer = CA_Hs_UI × s:CA_LE_UI-**set** **axiom** ∀ (_,s):CA_BE_Mer • **card** s=1
value
141. mereo_CA_BE: CA_BE → CA_BE_Mer

4.3.2.1.11.2 Confluence:

142. The mereology of a *canal confluence* is a pair: the unique identifier of the canal hub set it belongs and set of two or more canal element unique identfiers, one for each canal link incident upon the canal confluence.

type
142. CA_CO_Mer = CA_Hs_UI × s:CL_E_UI-**set** **axiom** ∀ (_,s):CA_CO_Mer • **card** s≥2
value
142. mereo_CA_CO: CA_CO → mereo_CA_CO

4.3.2.1.11.3 Outlet:

143. The mereology of an *outlet* is a pair: the unique identifier of the canal hub set it belongs and the singleton set of the unique identifier of the last canal link element for which it is the outlet.

type
143. CA_OU_Mer = CA_Hs_UI × s:CL_E_UI-**set** **axiom** ∀ (_,s):CA_OU_Mer • **card** s=1
value
143. mereo_CA_OU: CA_OU → CA_OU_Mer

4.3.2.1.12 Canal Links:

144. The mereology of a *canal link* are triples: the unique identifier of the canal link set to which it belongs, a two element set of the canal hubs that the link is linking, and a list (i.e., an ordered sequence) of the unique identifiers of the one or more canal link elements of the link.

type
144. CA_L_Mer = CA_Ls_UI × CA_H_UI-**set** × s:CA_LE_UI*
144. **axiom** ∀ (_,s,l):CA_L_Mer • **card** s=2 ∧ **len** l ≥ 1
value
144. mereo_CA_L: CA_L → CA_L_Mer

4.3.2.1.13 Sets of Canal Link Elements:

145. The mereology of any *canal link element* includes a pair: the unique identifier of the canal link to which it belongs and a two element set, one element is the unique identifier of either a canal hub or a[another] canal link element, the second element is the unique identifier of either a [next] canal link element or a canal hub – these we call CLE_UI_P.

type
145. CA_LE_Mer_Common = CL_UI × seuis:(CA_H_UI|CA_LE_UI)-**set**
145. **axiom** ∀ (clui,chluis):CA_LE_Mer • **card** chluis = 2

4.3.2.1.14 Canal Link Elements:

- Canal link elements are not individually "recognisable" as such. They are either canal channels, canal locks, canal locks with pumps or are canal polder pumps; cf. Item 64 on page 21.

4.3.2.1.14.1 Canal Channels:

146. The mereology of any *canal channel* is as the mereology included in any canal element mereology, cf. Item 145.

type
146. CA_CH_Mer = se:CA_LE_Mer_Common
value
146. mereo_CA_CH: CA_CH → CA_CH_Mer

4.3.2.1.14.2 Canal Locks:

147. The mereology of any *canal lock* is as the mereology included in any canal element mereology, cf. Item 145.

type
147. CA_LO_Mer = CA_LE_Mer_Common
value
147. mereo_CA_LO: CA_LO → CA_LO_Mer

4.3.2.1.14.3 Canal Polder Pumps:

148. The mereology of any *canal polder pump*, is a pair: in addition to the mereology of any canal link element – which is now first element of the pair, has the second element being the unique identifier of a polder.

type
148. CA_PO_PU_Mer = CA_LE_Mer_Common × P_UI
value
148. mereo_CA_PO_PU: CA_PO_PU → CA_PO_PU_Mer

4.3.2.2 The Mereology Axiom:

It is You, the domain analysers & describers, who decide on the mereologies of a domain ! You may wish to emphasize topological aspects of a domain; or you may wish to emphasize "co-ordination" relations between topologically "unrelatable" parts; or you may choose a mix of these; it all, also, depends on which aspects You wish to emphasize when transcendentally deducing [certain] parts into behaviours. Therefore the **mereology axiom** to be expressed reflects Your choice. Here we have chosen to emphasize the topological aspects of the canal domain. We use the term *well-formedness* of the mereology of an endurant. But do not be mislead ! It is not a property that we impose on the domain endurant. It is a fact. We cannot escape from that fact. Later, in the requirements engineering of a possible software product for a domain, You may decide to implement data structures to reflect mereologies, in which case you shall undoubtedly need to prove that your choice of data structures, their initialisation and update does indeed satisfy the axioms of the domain model.

149. For a canal system to be mereologically, cum topologically well-formed means that the canal system mereology is well-formed.

axiom
149. is_wf_CS_Mereology(cs_{end})

4.3.2.3 Well-formed Mereologies:

4.3.2.3.1 Canal Systems:

150. Canal system well-formednes, is_wf_CS_Mereology,
    (a) besides the appropriateness of its own mereology,
    (b) is secured by the well-formedness of the canal net aggregate and polder aggregate, is_wf_CN_Mereology and is_wf_PA_Mereology.

value
150. is_wf_CS_Mereology: CS → **Bool**
150. is_wf_CS_Mereology(cs_{end}) ≡
150a. **let** (cn_ui,pa_ui) = mereo_CS(cs_{end}) **in** [ι 132 π 30]
150a. cn_ui = cn_{uid} ∧ pa_ui = pa_{uid} **end** ∧ [ι 110 π 29, ι 111 π 29]
150b. is_wf_CN_Mereology(cn_{end})∧is_wf_PA_Mereology(pa_{end})

4.3.2.3.2 Canal Nets:

151. Well-formedness of canal nets, is_wf_CN_Mereology,
    (a) besides the appropriateness of its own mereology, is_wf_CS_Mereology,
    (b) is secured by the well-formedness of link and the hub aggregates, is_wf_CA_LA_Mereology, and all links, is_wf_CA_HA_Mereology.

value
151. is_wf_CN_Mereology: CN $\rightarrow$ **Bool**
axiom
151. is_wf_CN_Mereology(cn_{end}) $\equiv$
151a. **let** (cn_ui,ca_ha_ui,ca_la_ui) = mereo_CN(cn_{end}) **in** [ι 133 π 31]
151a. cn_ui = cn_{uid} $\wedge$ ca_ha_ui = ca_ha_{uid} $\wedge$ ca_la_ui = ca_la_{uid} **end** $\wedge$[ι 110 π 29, ι 112 π 29, ι 113 π 29]
151b. is_wf_CA_HA_Mereology(ca_ha_{end})$\wedge$is_wf_CA_LA_Mereology(ca_la_{end})

4.3.2.3.3 Polder Aggregates:

152. Well-formedness of polder aggregates, is_wf_PA_Mereology,
    (a) besides the appropriateness of its own mereology,
    (b) is secured by the well–formedness of the polder set is_wf_Ps_Mereology.

type
value
152. is_wf_PA_Mereology: PA $\rightarrow$ **Bool**
152. is_wf_PA_Mereology(pa_{end}) $\equiv$
152a. **let** (cs_ui,ps_ui) = mereo_PA(pa_{end}) **in** [ι 134 π 31]
152a. cs_ui = cs_{uid} $\wedge$ ps_ui = ps_{uid} **end** $\wedge$[ι 109 π 29, ι 114 π 29]
152b. is_wf_Ps_Mereology(ps_{end})

4.3.2.3.4 Canal Hub Aggregates:

153. Well-formedness of canal hub aggregates, is_wf_CA_HA_Mereology,
    (a) besides the appropriateness of its own mereology,
    (b) is secured by the well-formedness of its set of hubs.

value
153. is_wf_CA_HA_Mereology: CA_HA $\rightarrow$ **Bool**
153. is_wf_CA_HA_Mereology(hub) $\equiv$
153a. **let** (cnui,cahsui) = mereo_CA_HA(hub) **in** [ι 135 π 31]
153a. cnui = cn_{uid} $\wedge$ cahsui = ca_hs_{uid} **end** $\wedge$[ι 110 π 29, ι 115 π 29]
153b. is_wf_CA_Hs(ca_hs_{end})

4.3.2.3.5 Canal Link Aggregates:

154. Well-formedness of canal hub aggregates, is_wf_CA_HA_Mereology,
    (a) besides the appropriateness of its own mereology,
    (b) is secured by the well-formedness of its set of links.

value
154. is_wf_CA_LA_Mereology: CA_HA → **Bool**
154. is_wf_CA_LA_Mereology(la) ≡
154a. **let** (cnui,clsui) = mereo_CA_LA(la) **in** [ι 136 π 31]
154a. cnui = cn_{uid} ∧ clsui = cls_{uid} **end** ∧ [ι 110 π 29, ι 116 π 29]
154b. is_wf_CA_Ls(cls_{end})

4.3.2.3.6 Sets of Polders:

155. Well-formedness of sets of polders, is_wf_Ps_Mereology,
    (a) besides the appropriateness of its own mereology,
    (b) is secured by the well-formedness of its individual polders.

value
155. is_wf_Ps_Mereology: Ps → **Bool**
155. is_wf_Ps_Mereology(ps) ≡
155a. **let** (paui,puis) = mereo_Ps(ps_{end}) **in** [ι 137 π 32]
155a. paui = ca_pa_{uid} ∧ puis = ca_pos_{uid} **end** ∧ [ι 110 π 29, ι 111 π 29]
155b. ∀ po:PO • po ∈ pos_{end} ⇒ is_wf_P(po)

4.3.2.3.7 Sets of Hubs:

156. Well-formedness of a hub set is_wf_CA_Hs_Mereology,
    (a) besides the appropriateness of its own mereology,
    (b) is secured by the well-formedness of its individual hubs.

value
156. is_wf_CA_Hs_Mereology: CA_Hs → **Bool**
156. is_wf_CA_Hs_Mereology(ca_hs_{end}) ≡
156. **let** (cahaui,cahuis) = mereo_CA_Hs($ca\_bes_{end} \cup ca\_cos_{end} \cup ca\_ous_{end}$) **in** [$\iota$ 138 π 32]
156. cahaui = ca_ha_{uid} ∧ cahuis = ca_hs_{uid} **end** ∧ [ι 112 π 29, ι 115 π 29]
156a. ∀ hub:CA_H • hub ∈ ca_hs_{end} ⇒ is_wf_CA_H(hub)

4.3.2.3.8 Sets of Links:

157. Well-formedness of sets of links

    (a) besides the appropriateness of its own mereology,

    (b) is secured by the well-formedness of all of its individual links.

value
160. is_wf_CA_Ls_Mereology: mereo_CA_Ls → **Bool**
160. is_wf_CA_Ls_Mereology(ca_ls_{end}) ≡
160a. **let** (cs_la_ui,ca_ls_ui) = mereo_CA_Ls(ca_ls_{end}) **in** [ι 139 π 32]
160a. cs_la_ui = ∧ ca_ls_ui = ca_ls_{uid} **end** ∧ [ι 113 π 29, ι 116 π 29]
160b. ∀ link:CA_L • link ∈ ca_ls_{end} ⇒ is_wf_CA_L(link)

4.3.2.3.9 Polders:

158. Well-formedness of polders, is_wf_P_Mereology, depends jst on the appropriateness of its own mereology.

value
158. is_wf_Mereology_Polder: mereo_P → **Bool**
158. is_wf_Mereology_Polder(p) ≡
158. **let** ps_ui = mereo_P(p) **in** [ι 140 π 32]
158. ps_ui = ps_{uid} **end** ≡ [ι 114 π 29]

4.3.2.3.10 Hubs:

58 Hubs are not individually "recognisable" as such. They are either begin/ends, confluences or outlets; cf. Item 58 on page 21.

159. The well-formedness of hubs thus "translates" into the well-formedness of either begin/ends, confluences or outlets.

type
58. CA_H == CA_BE|CA_CO|CA_OU
value
159. is_wf_Mereology_H: H → **Bool**
159. is_wf_Mereology_H(h) ≡
159. is_CA_BE(h) → is_wf_Mereology_CA_BE(h),
159. is_CA_CO(h) → is_wf_Mereology_CA_CO(h),
159. _ → is_wf_Mereology_CA_OU(h)

4.3.2.3.10.1 Begin/End:

160. Well-formedness of sets of links

    (a) besides the appropriateness of its own mereology,

(b) is secured by the well-formedness of all of its individual links.

161. Well-formedness of the mereology of begin/end hubs, is_wf_Mereology_CA_BE, depends just on the appropriateness of their own mereology.

value
161. is_wf_Mereology_CA_BE: CA_BE → **Bool**
161. is_wf_Mereology_CA_BE(be) ≡ ≡
161. **let** (cahsui,cleuis) = mereo_CA_BE(be) **in** [ι 141 π 33]
161. cahsui ∈ ca_hs_{uid} ∧ cleuis ∈ ca_les_{uid} **end** [ι 115 π 29, ι 121 π 30]

4.3.2.3.10.2 Confluence:

162. Well-formedness of the mereology of confluence hubs, is_wf_Mereology_CA_CO, depends just on the appropriateness of their own mereology.

value
162. is_wf_Mereology_CA_CO: CA_CO → **Bool**
162. is_wf_Mereology_CA_CO(co) ≡
162. **let** (cahsui,cleuis) = mereo_CA_CO(co) **in** [ι 142 π 33]
162. cahsui ∈ ca_hs_{uid} ∧ cleuis ∈ ca_les_{uid} **end** [ι 115 π 29, ι 121 π 30]

4.3.2.3.10.3 Outlet:

163. Well-formedness of the mereology of outlet hubs, is_wf_Mereology_CA_OU, depends just on the appropriateness of their own mereology.

163. is_wf_Mereology_CA_OU: CA_CO → **Bool**
163. is_wf_Mereology_CA_OU(ou) ≡
163. **let** (cahsui,cleuis) = mereo_CA_OU(ou) **in** [ι 143 π 33]
163. cahsui ∈ ca_hs_{uid} ∧ cleuis ∈ ca_les_{uid} **end** [ι 115 π 29, ι 121 π 30]

4.3.2.3.11 Canal Links:

164. The well-formedness of canal links depends on

(a) the appropriateness of its own mereology, that is, that its unique identifier references are indeed to canal system identifiers,

(b) the well-formedness of the set of link elements that can be observed from a canal link, that is, that they form a sequence of canal link elements – connecting two canal hubs, and

(c) the ("remaining") well-formedness of the canal link elements.

164. is_wf_Mereology_CA_L: CA_L → **Bool**
164. is_wf_Mereology_CA_L(link) ≡
164a. **let** (calsui,cahuis,caleuil) = mereo_CAL_L(link) **in** [ι 143 π 33]
164a. calsui = ∧ cahuis = ∧ caleuil = ∧ [ι 115 π 29, ι 121 π 30, ι 121 π 30]
164b. wf_Link_Es(obs_CA_LEs(link))(cahuis)(caleuil) ∧
164c. ∀ le:CA_LE•le ∈ obs_CA_LEs(link) ⇒ is_wf_Mereology_CA_LE(le) **end**

4.3.2.3.12 Well-formed Sets of Canal Link Elements: The introduction of the wf_Link_Es predicate represents a slight deviation from the introduction of the usual is_wf_Mereology predicates.

165. The wf_Link_Es predicate applies to a set of link elements, link, and a unique identifier list, uil, of unique link element identifiers. The predicate holds if the set, link: CA_LE-**set**, of link elements not only can be ordered in the sequence indicated by uil.
    (a) The length of the unique identifier list, uil, must match the cadinality of the set link.
    (b) Let linkl be the list of link elements prescribed by uil.
    (c) Now there are the following cases of "neighbour" mereologies to observe:
        i. For a singleton list, linkl, its only element must connect the two distinct hubs identified in cahuis.
        ii. for a two-element unique identifier list, ⟨luil,ruil⟩ one of their common mereology identifiers are shared, i.e., their elements are connected, and the other common mereology identifiers are those of canal hubs, i.e. end-points.
        iii. For lists of length three or more elements
            A. the first and last elements must have end-points,
            B. and for all elements in-between it must be the case that the neighbour identifiers
            C. of the previous and the following link elements
            D. must share identifiers with the quantified element
            E. and share identifier with

```
value
165.        wf_Link_Es: CA_LE-set → CA_H_UI-set × CA_LE_UI* → Bool
165.        wf_Link_Es(link)(euis:{l_ca_h_ui,r_ca_h_ui})(uil) ≡ [ axiom card euis = 2 ]
165a.           card link = len uil ∧
165b.           let linkl = ⟨ le | i:Nat, ce:C_LE•
165b.                                1≤i≤len uil∧le∈link∧uid_CA_LE(le)=uil[i]⟩ in
165c.           case linkl of
165(c)i.            ⟨ui⟩ →
165(c)i.                cahauis = seuis(mereo_LE(get_part(ui))),
165(c)i.                axiom: let {lui,rui}=seuis(mereo_LE(get_part(ui))) in
165(c)i.                           wf_end_points(lui,rui)(euis) end
165(c)ii.           ⟨lui,rui⟩ →
165(c)ii.               wf_end_points(lui,rui)(euis),
165(c)iii.          ⟨lui⟩⌢link⌢⟨rui⟩ → [ axiom: len linkl ≥ 3, i.e., link≠⟨⟩ ]
165(c)iiiA.             wf_end_points(lui,rui)(euis) ∧
165(c)iiiB.             ∀ i:Nat • 1<i<len linkl ⇒
165(c)iiiC.                 let {uim1,uim1}=seuis(mereo_CA_LE(get_part(linkl[i−1]))),
165(c)iiiC.                     {uip1,uip1}=seuis(mereo_CA_LE(get_part(linkl[i+1]))) in
165(c)iiiD.                 axiom: lui∈{uim1,uim1} ∧ rui∈{uip1,uip1}
165(c)iiiE.                 let uism1={uim1,uim1}\{lui}, uisp1={uip1,uip1}\{rui} in
165(c)iiiE.                 link[i] = uism1 = uisp1 end
165.        end end  end

165(c)ii.   is_shared: UI × UI-set × UI-set → Book
```

165(c)ii. is_shared(ui,luis,ruis) $\equiv$ ui $\in$ luis $\cap$ ruis
165(c)ii.
165(c)ii. shared: **UI-set** $\times$ **UI-set** $\rightarrow$ UI
165(c)ii. shared(luis,ruis) $\equiv$ luis $\cap$ ruis
165(c)ii. **pre**: $\exists$ ui:UI • is_shared(ui,luis,ruis)
165(c)ii.
165(c)ii. wf_end_points: (UI$\times$UI) $\rightarrow$ CA_H_UI-**set** $\rightarrow$ **Bool**
165(c)ii. wf_end_points(lui,rui)(euis) $\equiv$ [**axiom**: **card** euis = 2]
165(c)ii. **let** {llui,lrui} = seuis(mereo_LE(get_part(lui))),
165(c)ii. {rlui,rrui} = seuis(mereo_LE(get_part(rui))) **in**
165(c)ii. **if** $\exists$ ui:CA_LE_UI • is_shared(ui,{llui,lrui},{rlui,rrui})
165(c)ii. **then let** ui = shared({llui,lrui},{rlui,rrui}) **in**
165(c)ii. {llui,lrui,rlui,rrui}\{ui}=euis $\wedge$
165(c)ii. euis$\subseteq$$ca\_bes_{uid}$$\cup$$ca\_cus_{uid}$$\cup$$ca_ous_{uid}$ **end**
165(c)ii. **else false end end**,

4.3.2.3.13 Canal Link Elements: ...

4.3.2.3.13.1 Canal Channels:

166. is_wf_CA_CH_Mereology,

    (a)

    (b)

    (c)

??.
??.
166a.
166b.
166c.

4.3.2.3.13.2 Canal Locks:

167. is_wf_CA_LO_Mereology,

    (a)

    (b)

    (c)

167.
167.
167a.
167b.
167c.

4.3.2.3.13.4 Canal Polder Pumps:

168. (a)
     (b)
     (c)

168.
168.
168a.
168b.
168c.

4.3.3 Canals: Routes

4.3.3.1 Preliminaries:

169. By an end-identifier we mean the unique identifier of a begin/end or an outlet.

170. By a middle-identifier we mean the unique identifier of a confluence, channel, lock, lock with pump or a polder pump.

171. By a unit identifier we mean either an end-identifier or a middle-identifier.

172. By a canal route we mean a sequence of one or more unique identifiers of atomic canal entities, two if one of the identifiers is that of a begin/end or an outlet unit.

Notice that adjacent canal route identifiers be distinct. But a triplet of adjacent canal route identifiers may have the same first and last elements. It is allowed that a route, so-to-speak, goes forward and backward. There is, in a sense, no preferred directions in canal systems.

type
169. E_UI = CA_BE_UI|CA_OU_UI
170. M_UI = CA_CO_UI|CA_CH_UI|CA_LO_UI|CA_LO_PU_UI|CA_PO_PU_UI
171. UI = E_UI|M_UI
172. CR = UI*
axiom
172. $\forall$ cr:CR,i:**Nat** • $\{$i,i+1$\}\subseteq$**inds** cr $\Rightarrow$ cr[i]$\neq$cr[i+1]

173. Let uid_MU be a "common" unique identifier observer of middle units.

174. Let mereo_MU be a "common" mereology observer of middle units other than polder pumps.

175. From middle units, i.e., confluences, channels, locks, lock with pumps and polder pumps we can extract simple, one-, two- or three element canal routes.

```
type
173. uid_MU = uid_CA_CO|uid_CA_CH|uid_CA_LO|uid_CA_LO_PU|uid_CA_PO_PU
174. mereo_MU = mereo_CA_CO|mereo_CA_CH|mereo_CA_LO|uid_CA_LO_PU
175. MU = CA_CO|CA_CH|CA_LO|CA_LO_PU|CA_PO_PU
value
175. xtr_M_UI_CRs: MU → CR-set
175. xtr_M_UI_CRs(mu) ≡
175.     let mu_ui = uid_MU(mu),
175.         {ui1,ui2} =
175.             is_CA_PO_PU(mu) →
175.                 let (_,cuis,_) = mereo_CA_PO_PU(mu) in cuis end
175.             _ → let (_,cuis) = mereo_MU(mu) in cuis end
175.     {⟨mu_ui⟩,⟨ui1,mu_ui⟩,⟨ui2,mu_ui⟩,⟨mu_ui,ui1⟩,⟨mu_ui,ui2⟩,⟨ui1,mu_ui,ui2⟩,⟨ui2,mu_ui,ui1⟩}
175.     end
```

4.3.3.2 Routes:

176. By means of xtr_M_UI_CRs we can extract, xtr_CRs(mus), the infinite set of canal routes from any set, mus, of middle canal elements.

177. First we calculate initial, i.e., simple routes.

178. Then for every two routes, a “left” and a “right” route, in the set of routes beng recursively defined, such that the last element of the left route is identical to the first element of the right route, the route formed by concatenating the left and right routes “around” the shared element is a route.

179. The set of routes of a canal system is the least fix-point soluion the the equation of Item 178.

180. No two adjacent identifiers are the same.

```
type
175. MU = CA_CO|CA_CH|CA_LO|CA_LO_PU|CA_PO_PU
value
176. xtr_CRs: MU-set → CR-infset
176. xtr_CRs(mus) ≡
177.     let icrs = ∪{xtr_M_UI_CRs(mu)|mu:MU•mu ∈ mus} in
178.     let crs = icrs ∪ {lr^⟨ui⟩^rl|lr,⟨ui⟩,rr:CR•{lr^⟨ui⟩,⟨ui⟩^rr}∈crs} in
179.     crs
180.     axiom: ∀ cr:CR, i:Nat • cr isn crs ∧ {i,i+1}⊆inds cr ⇒ cr[i]≠cr[i+1]
176.     end end
```

4.3.3.3 Connected Canal Systems:

181. Canal systems, such as we shall understand them, are connected.

182. That is, there is a route from any canal element to any other other canal element.

183. Let mus be the set of all middle elements of a canal system.

184. Let rs be the infinite set of all routes of mus.

185. Now, for any two unique identifiers of middle elements there must be a route in rs.

value
181. is_connected_CS: CS $\rightarrow$ **Bool**
182. is_connected_CS(cs) $\equiv$ **in**
183. **let** mus $= ca\_cos_{end} \cup ca\_chs_{end} \cup ca\_los_{end} \cup ca\_lo\_{}_{p}us_{end} \cup ca\_po\_pus_{end}$ **in**
184. **let** rs = xtr_CRs(mus) **in**
185. $\forall$ ui:M_UI $\cdot$ $\{$fui,tui$\}$ $\subseteq$ uid_MU(mus) $\Rightarrow$ $\exists$ r:R $\cdot$ r $\in$ rs and r$[1]$=r$[$**len** r$]$
182. **end end**

4.3.3.4 A Canal System Axiom:

186. Canal systems are connected.

axiom
186. $\forall$ cs:CS $\cdot$ is_connected_CS(cs)

4.3.4 Canals: Attributes

We shall treat the issue of canal part attributes, not, as is usual, one-by-one, sort-by-sort, but more-or-less "collectively", across canal hubs and links. And we do so category-by-category of attribute kinds: spatial, temporal and other.

4.3.4.1 Spatial and Temporal Attributes:

4.3.4.1.1 Spatial Attributes: Natural and artefactual, that is, mane-made endurants reside in space. We have dealt with space, i.e., $\mathbb{SPACE}$, in [1, *Sects. 2.2 and 3.4*]. Subsidiary spatial concepts are those of $\mathbb{VOLUME}$, $\mathbb{AREA}$, $\mathbb{CURVE}$ (or $\mathbb{LINE}$), and $\mathbb{POINT}$. All canal system endurants possess, whether we choose to model them or not, such spatial attributes. We shall not here be bothered by any representation, let alone computational representations, of spatial attributes. They are facts. Any properties that two $\mathbb{AREA}$s, a_i and a_j may have in common – like **bordering**, **overlapping disjoint** or **properly contained** – are facts and should, as such be expressed in terms of **axiom**s. They are not properties that can, hence must, be proven. Once a *domain description*, involving spatial concepts is the base for a *requirements prescription*, then, if these spatial concepts are not *projected* out of the evolving requirements, they must, eventually, be prescribed – or assumed to have – computable representations. In that case axioms concerning spatial quantities are turned into **proof obligations** that must, eventually, be **discharged**.

Let us establish the following spatial attributes, common to all canal parts:

187. Location: A single $\mathbb{POINT}$ in $\mathbb{SPACE}$ characterised by its longitude, latitude and altitude, the latter height above or depth below sea level, including 0. How these are measured is of no concern in this model.

188. Extent: An $\mathbb{AREA}$, i.e., a plane in $\mathbb{SPACE}$, i.e., a dense set of $\mathbb{POINT}$s according to some topology.

189. Volume: A proper subset $\mathbb{SPACE}$, i.e., a three dimensional dense set of $\mathbb{POINT}$s $\mathbb{SPACE}$, according to some topology.

190. The Location of a canal part is always **embedded** in its Extent.

191. The Extent of a canal part is always **embedded** in its Volume

type
187. Location
188. Extent
189. Volume
value
187. attr_Location: CS|CN|PA|CA_HA|CA_LA|CA_LE → Location
188. attr_Extent: CS|CN|PA|CA_HA|CA_LA|CA_LE → Extent
189. attr_Volume: CS|CN|PA|CA_HA|CA_LA|CA_LE → Volume
189. **is_embedded**: Location × Extent → **Bool**, **is_embedded**: Extent × Volume → **Bool**
axiom
190. ∀ e:(CS|CN|PA|CA_HA|CA_LA|CA_LE)•**is_embedded**(attr_Location(e),attr_Extent(e))
191. ∀ e:(CS|CN|PA|CA_HA|CA_LA|CA_LE)•**is_embedded**(attr_Extent(e),attr_Volume(e))

Let us establish the following ***s, common to some canal parts:

192. Let us addume the sort notions of Latitude, Longitude and Altitude,

193. And let us assume "sea level" Altitude value "0".

194. A projected extent is an extent all of whose **altiude** elements are zero (0), i.e., "at sea level".

195. We assume functions, latitude, logitude, altitude, that extract respective elements of a point.

196. No two distinct hubs and link elements can share neither location, area nor volume – so they are **disjoint**.

Canal channels may share projected extents.

type
192. Latitude, Longitude, Altitude
value
193. ″0″: Altitude
194. projected_Extent: Extent → Extent
195. latitude: $\mathbb{POINT}$ → Latitude, longitude: $\mathbb{POINT}$ → Longitude, altitude: $\mathbb{POINT}$ → Altitude
axiom
196. ∀ e,e′:(CS|CN|PA|CA_HA|CA_LA|CA_LE): e≠e′ ⇒ **disjoint**(attr_Volume(e),attr_Volume(e′))
192. ∀ e,e′:CA_CH •
193.
194.

4.3.4.1.2 Temporal Attributes: Natural and artefactual, that is, mane-made endurants reside in time. We have dealt with space, i.e., $\mathbb{TIME}$, in [1, *Sects. 2.2 and 3.5*]. Subsidiary spatial concepts are those of $\mathbb{TIME}$ and $\mathbb{TIME}$ $\mathbb{INTERVAL}$s. All canal system endurants possess, whether we choose to model them or not, such temporal attributes. We shall not here be bothered by any representation, let alone computational representations, of temporal attributes. They are facts. Any properties that two $\mathbb{TIME}$ $\mathbb{INTERVAL}$s, ti_i and ti_j may have in common, like **bordering** or **overlapping**, are facts and should, as such be expressed in terms of **axiom**s[25]. They are not properties that can, hence must, be proven. Once a *domain description*, involving temporal concepts is the base for a *requirements prescription*, then, if these temporal concepts are not *projected* out of the evolving requirements, they must, eventually, be prescribed – or assumed to have – computable representations. In that case axioms concerning temporal quantities are turned into **proof obligations** that must, eventually, be **discharged**.

4.3.4.1.3 Event Attributes: Some events can, for example, be talked about, by humans. They, so-to-speak, belong to an event-category: *"von hörensagen"*. Examples are: *"a canal lock opened at time τ "*; *"a polder pump stopped pumping at time τ' "*; and *"a canal vessel passed a certain canal channel point at time τ'' "*. Let refer to the event as e:E. If, for an endurant, p of sort P, they are relevant to an analysis & description of a domain, then they must be noted, for example in the form of an attribute named, say, history_E:

type history_E = $\mathbb{TIME} \xrightarrow[m]{} E$

4.3.4.1.4 Continuous Time Attributes: Mostly one models discrete time phenomena. But often phenomena are continuous time varying. Examples are: *"the canal water level"*, *"the canal water temperature"*, and *"the position of a vessel along a canal"*. If, for an endurant, p of sort P, such a phenomenon, e:E, is relevant to an analysis & description of a domain, then it must be noted, for example in the form of an attribute named, say, history_E:

type history_E = $\mathbb{TIME} \rightarrow E$

4.3.4.2 Canal System, Net and Polder Attributes:

197. Canal systems have location and extent.
198. So do canal nets and
199. polder aggregates.
200. Canal nets and polder aggregates are **bordering**[26].
201. Canal nets and polder aggregates are properly **embedded**[27] in canal systems.
202. Etc.

[25] We refer here to the $\mathbb{TIME}$ and $\mathbb{TIME}$ $\mathbb{INTERVAL}$ operators of [1, *Sects. 2.2 and 3.5*]

[26] We leave it to a chosen Topology to define the **are_bordering** predicate

[27] We leave it to a chosen Topology to define the **is_embedded** predicate

value
197. attr_Location: CS $\rightarrow$ Location; attr_Extent: CS $\rightarrow$ Extent
198. attr_Location: CN $\rightarrow$ Location; attr_Extent: CN $\rightarrow$ Extent
199. attr_Location: PA $\rightarrow$ Location; attr_Extent: PA $\rightarrow$ Extent
axiom
200. $\forall$ cs:CS • **are_bordering**(attr_Extent(obs_CN(cs)),attr_Extent(obs_PA(cs)))
201. $\forall$ cs:CS • **is_embedded**(attr_Extent(obs_CN(cs)),cs)$\wedge$**is_embedded**(cs,attr_Extent(obs_PA(cs)))
202. ...

4.3.4.3 Canal Hub and Link Attributes: Two kinds of attributes shared across hubs and links, therefore their elements, stand out: *water level*s and *ambient* and *water temperature*s.

4.3.4.3.1 Water Temperatures:

203. Let there be given a notion of water temperature.

Generally, over time, one can associate with any canal hub and link element,

204. high,
205. normal and
206. low water

water temperatures, and specifically, at any time,

207. current water temperatures.

type
245. Wa_Temp
246. Hi_Temp = $\mathbb{TIME}$ $\rightarrow$ Wa_Temp
247. No_Temp = $\mathbb{TIME}$ $\rightarrow$ Wa_Temp
248. Lo_Temp = $\mathbb{TIME}$ $\rightarrow$ Wa_Temp
249. Cu_Temp = Wa_Temp
value
246. attr_Hi_Temp: H $\rightarrow$ Hi_Temp, attr_LE_Temp: LE $\rightarrow$ Hi_Temp
247. attr_No_Temp: H $\rightarrow$ No_Temp, attr_LE_Temp: LE $\rightarrow$ No_Temp
248. attr_Lo_Temp: H $\rightarrow$ Lo_Temp, attr_LE_Temp: LE $\rightarrow$ Lo_Temp
249. attr_Cu_Temp: H $\rightarrow$ Cu_Temp, attr_LE_Temp: LE $\rightarrow$ Cu_Temp

The Hi_Temp, No_Temp and Lo_Temp attributes are normally continuous functions over time. They are facts. One does not have to "go out" and measure them ! We do not have to think of representations for the Hi_Temp, No_Temp and Lo_Temp attributes.

4.3.4.3.2 Water Levels:

208. Let there be given a notion of water level.

Generally, over time, one can associate with any canal hub and link element,

209. high,
210. normal and
211. low

water levels, and specifically, at any time,

212. current water level.

type
208. Wa_Lev
209. Hi_WL = $\mathbb{TIME}$ → Wa_Lev
210. No_WL = $\mathbb{TIME}$ → Wa_Lev
211. Lo_WL = $\mathbb{TIME}$ → Wa_Lev
212. Cu_WL = Wa_Lev
value
209. attr_Hi_WL: H → Hi_WL, attr_LE_WL: LE → Hi_WL
210. attr_No_WL: H → No_WL, attr_LE_WL: LE → Hi_WL
211. attr_Lo_WL: H → Lo_WL, attr_LE_WL: LE → Hi_WL
212. attr_Cu_WL: H → Cu_WL, attr_LE_WL: LE → Hi_WL

The Hi_WL, No_WL and Lo_WL attributes are normally continuous functions[28] over time. Remarks on Hi_WL, No_WL and Lo_WL attributes similar to those of the Hi_Temp, No_Temp and Lo_Temp attributes as to continuity and representations apply.

4.3.4.4 Canal Hub Attributes:

4.3.4.4.1 Canal Begin/End Attributes:

213.
214.
215.
216.
217.
218.
219.

[28]– barring cyclones, tornados and the like !

220.

213.
214.
215.
216.
217.
218.
219.
220.

4.3.4.4.2 Canal Confluence Attributes:

221.

222.

223.

224.

225.

226.

227.

228.

221.
222.
223.
224.
225.
226.
227.
228.

4.3.4.4.3 Canal Outlet Attributes:

229.

230.

231.

232.

233.

234.

235.

236.

229.
230.
231.
232.
233.
234.
235.
236.

4.3.4.5 Canal Links Attribute:

237.

238.

239.

240.

241.

242.

243.

244.

237.
238.
239.
240.
241.
242.
243.
244.

4.3.4.5.1 Canal Channel Attributes:

245.

246.

247.

248.

249.

250.

251.

252.

245.
246.
247.
248.
249.
250.
251.
252.

4.3.4.5.2 Canal Lock Attributes:

253.

254.

255.

256.

257.

258.

259.

260.

253.
254.
255.
256.
257.
258.
259.
260.

4.3.4.5.3 Canal Polder Pump Attributes:

261.

262.

263.

264.

265.

266.

267.

268.

261.
262.
263.
264.
265.
266.
267.
268.

4.3.4.6 Polder Attributes:

269.

270.

271.

272.

273.

274.

275.

276.

269.
270.
271.
272.
273.
274.
275.
276.

4.3.5 Well-formedness of Attributes

277. There is a predicate, is_wf_CS_Attributes.
278.
279.
280.
281.

```
277. is_wf_CS_Attributes: CS → Bool
277. is_wf_CS_Attributes(cs) ≡ ...
278.
279.
280.
281.
```

4.4 Speculations

To be Written

5 Conclusion

To be Written

6 Bibliography

References

[1] Dines Bjørner. *Domain Science & Engineering, A Basis for Understanding Man-made Worlds – A Foundation for Software Development*. EATCS Monographs in Theoretical Computer Science. Springer, 2021.

A Terminologies

A.1 A Rivers Terminology

Almost all of the terms listed here are from www.primaryhomeworkhelp.co.uk/rivers/glossary.html.

Basin: (Drainage basin) The area of land that is drained by a river and its tributaries.

Canal: An artificial or man-made river channel.

Channel: A groove in the land that a river flows along.

Confluence: Where two or more rivers or streams meet.

Current: The flow of the river.

Dam: A barrier built, usually across a watercourse, for holding back water or diverting the flow of water.

Delta: A fan-shaped area of sediment built up at the mouth of a river.

Dock: A place for vessels to load and unload cargo or to be repaired.

Downstream: The direction that the river flows, towards the mouth of the river.

Drainage Basin: The area of land that is drained by a river and its tributaries. The boundary of a river basin is called the watershed.

Estuary: A drowned river valley in a coastal lowland area. Occurs near or at the mouth of a river, where the tide meets the current and the fresh and salt waters mix.

Fjord: A fjord is a deep, narrow flooded inlet of the sea that was formed during the last Ice Age.

Flood: Flooding happens when a river has too much water in its channel. The water breaks through the river banks and spreads over the surrounding land.

Flow: The direction of water flow.

Flood barrier: A barrier forming a temporary dam that may be erected quickly or permanently alongside a river to protect a flood-prone area.

Floodplain: The flood plain is the flat land of the river valley close to the river banks. The floodplain is usually found in the lower course of a river. It is a fertile area of land, used for agriculture and growing crops.

Ford: A point where a road goes through a river.

Freshwater: Water that has no salt in it.

Gorge: A gorge is a steep-sided river valley which is very narrow and deep. Most gorges have rocky sides. The river cuts this deep valley by erosion. Gorges are created over thousands of years.

Irrigation: The supply of water to farmland so that crops can grow in areas where water supplies are scarce or unreliable. In areas where there is not much rainfall, farmers irrigate the land, by diverting water from rivers to their fields, in channels, ditches or pipes.

Meander: A bend in a river - usually in the middle or lower course. The meander continually changes shape as the fast flowing current of water erodes the outside bank of the meander bend and deposition occurs in the slack water of the inside of the bend.

Mouth: The end of the river. The mouth may be where the river meets the sea, a lake or a larger waterway. Most rivers flow out into the sea, and this is where they end their journey.

Moorings: The place where a ship or boat is docked (or tied up).

Oxbow lake: A small arc-shaped lake formed when a meander is sealed off by deposition. Oxbows are only found on river floodplains.

Rapids: Rapids are fast-flowing stretches of water formed where the river surface breaks up into waves because rocks are near to the surface.

Ravine: Another name for a narrow gorge.

Reservoir: A reservoir is an artificial lake created by building a dam across a river.

River: A river is a naturally winding watercourse that drains surplus water from a drainage basin.

River channels: The trenches in which rivers flow for most of the year

Source: Where the stream begins: usually where there is a spring, and quite high up.

Spring: A place where water naturally seeps or gushes from the ground - often in marsh or bog areas.

Stagnant: No water flow.

Tributary: Stream or river that feeds into a larger watercourse.

Upstream: Opposite to the currents flow – towards the source of the river.

Watershed: High ground that surrounds a drainage basin. The boundary of a river basin.

Waterfall: A place where the river course is interrupted by a tall step.

Water table: The water table is the natural level of water in a soil or rock. Below the water table the soil or rock is saturated.

B Indexes

B.1 Canal Indexes

B.1.1 Canal Sorts and Types

B.1.2 Canal Predicates and Functions

B.1.3 Canal Values

B.1.4 Canal Axioms

Canal Systems Axioms

Unique Identifier Axioms

C All Indexed Canal Terms

There are 202 indexed terms.

D Further Canal and Polder Photos

D.1 Canal Photos

Figure 21: Middle: The Corinth Canal, Greece

D.2 Polder Photos

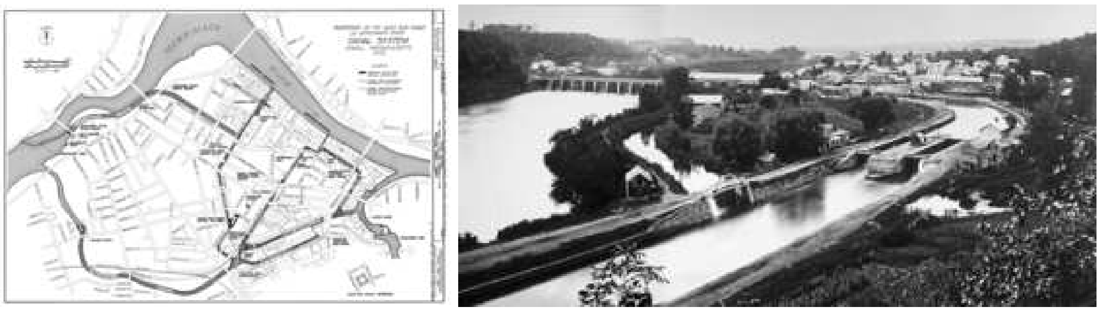

Figure 22: Left: Lowel Canals, Mass., USA

Figure 23: Left: The Panama Canal !

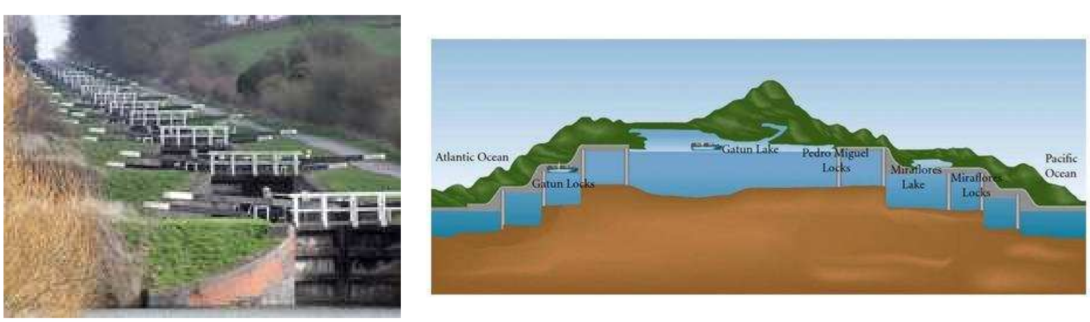

Figure 24: Left: 16 consecutive locks at Caen Hill, the Kennet and Avon Canal, Wiltshire, England

Figure 25:

Figure 26:

Figure 27:

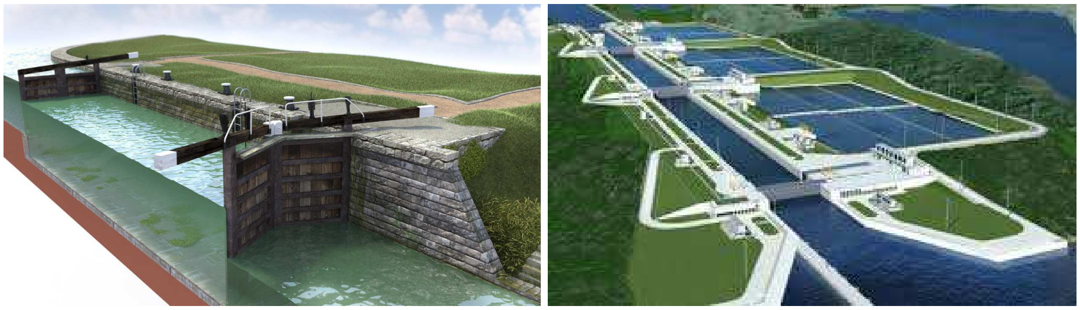

Figure 28:

Figure 29: The Trent-Severn Canal, England

Figure 30: The Suez Canal

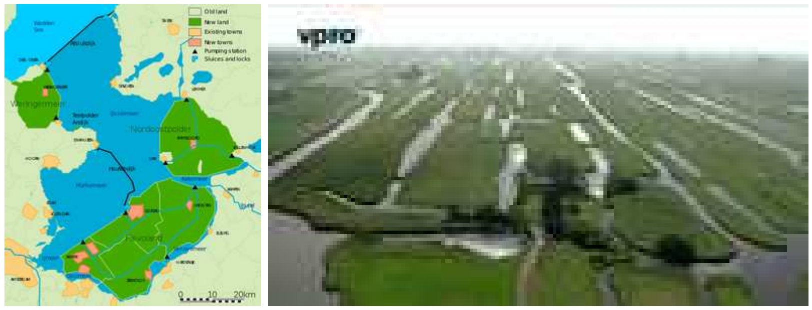

Figure 31:

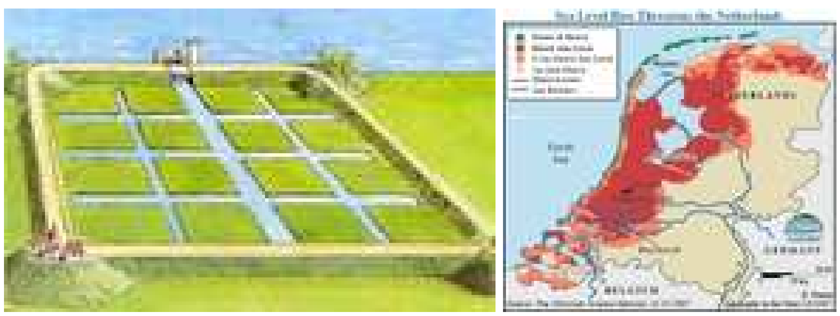

Figure 32:

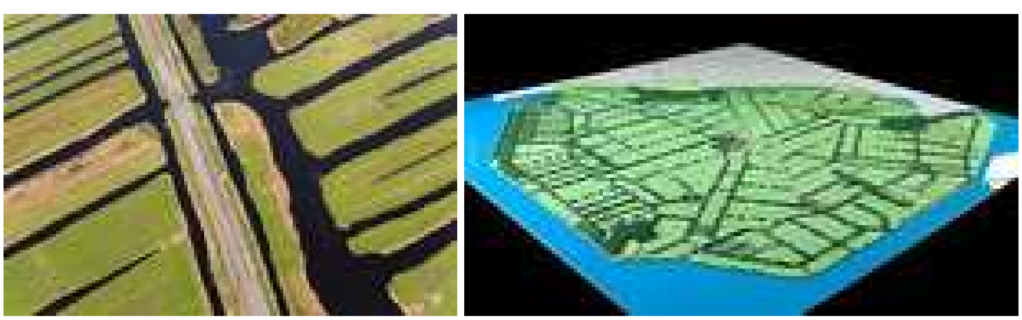

Figure 33:

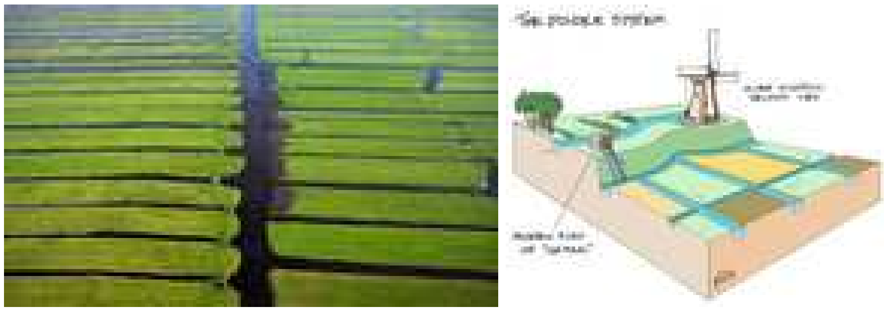

Figure 34:

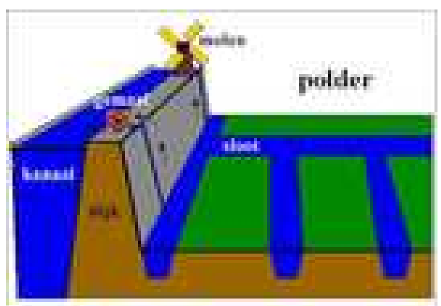

Figure 35:

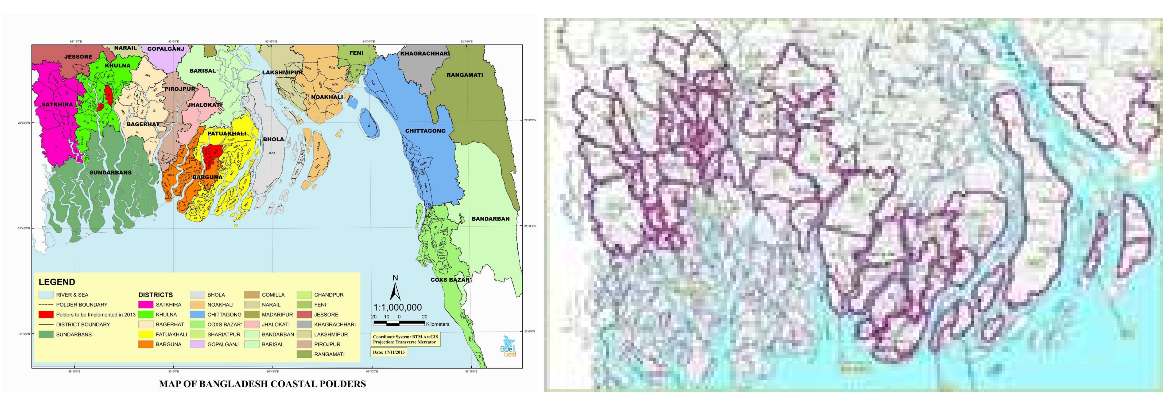

Figure 36: Bangladesh Polders

Figure 37: Polder Pump (NL) and River Lock (US)

Figure 38:

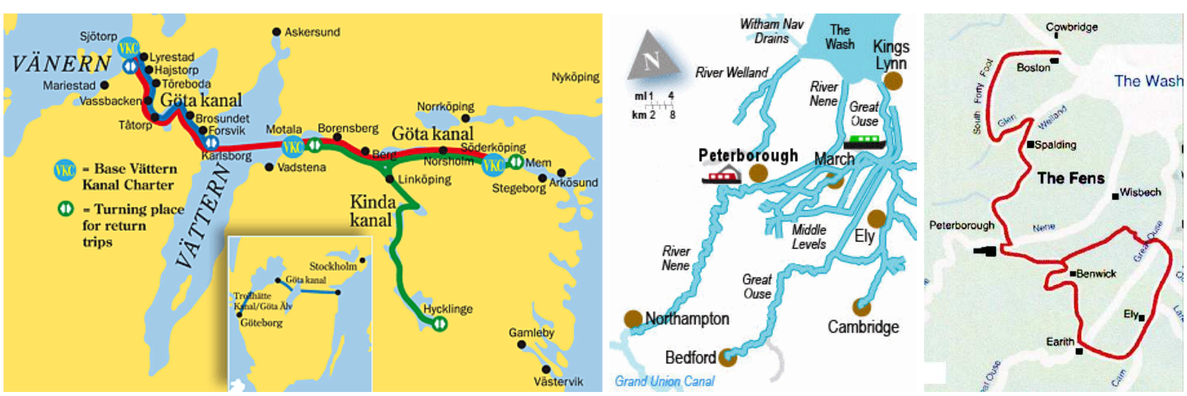

Figure 39: Göta Kanal. The Fens, England