# Navigocorpus: A Database for Shipping Information - A Methodological and Technical Introduction

Jean-Pierre Dedieu, Silvia Marzagalli, Pierrick Pourchasse, Werner Scheltens

# Navigocorpus: A Database for Shipping Information – A Methodological and Technical Introduction

**Jean-Pierre Dedieu, Silvia Marzagalli, Pierrick Pourchasse and Werner Scheltjens**

The post-punch-card generation of electronic data processing has opened exciting opportunities for historians interested in tracking trade and shipping. Within the past few decades, many scholars have patiently collected data and stored them in databases which are generally organized according to the nature of the sources used and the kind of questions they wish to pose. Once stored in a database, the information can be used both for micro- and macro-analyses.

Once their creators have exploited them to write essays and books, databases tend to be more-or-less forgotten. They are not always updated and sometimes not even preserved; hard disks occasionally crash; and software becomes obsolete. Perhaps most regrettably, the information they contain is not made available to colleagues or to the public. Still, the reality is that many scholars are willing to make their databases available to others once they have used them for their own projects.

These realities led us to decide to create an open, on-line database called Navigocorpus.[1] After several years of work, it will be possible to query

[1]Navigocorpus ("Corpus des itinéraires des navires de commerce, XVIIe-XIXe siècles" ("Database on the Itineraries of Merchant Ships, 17th-19th Centuries"), is a research project sponsored by the French Agence Nationale de la Recherche (ANR-07-CORP-028) and coordinated by Silvia Marzagalli (Centre de la Méditerranée Moderne et Contemporaine, Nice) in collaboration with Jean-Pierre Dedieu (Laboratoire de Recherche Historique Rhône-Alpes, Lyon) and Pierrick Pourchasse (Centre de Recherche Bretonne et Celtique, Brest). Navigocorpus has attracted the support of many colleagues who have generously put their databases at our disposal. We have also collected data ourselves from various sources to test the database design. Some already existing databases which were elaborated in collaborative projects will be incorporated into Navigocorpus in the coming years, such as those on eighteenth-century Greek shipping (created by Gelina Harlaftis); eighteenth- and nineteenth-century southern Italian trade with Marseille (Biagio Salvemini); and early modern Göteborg shipping (Christina Dahlede). In dealing with databases provided by colleagues we intend to treat them as meta-sources, accepting that those who created them might not have collected all the information in the sources. For our own database, however, we decided to collect all of the information so that a researcher would not need to return to the original source. For more information, see

the database on-line beginning in 2012. We acknowledge readily and gratefully the inspiration provided by the Trans-Atlantic Slave Trade Database.[2] Navigocorpus, however, is far more ambitious, as it potentially can deal with all kinds of ships (from a one-ton sailing vessel to a container ship) and trade regardless of location or time period. Although the project was funded for only four years, we hope that Navigocopus will grow over time. Indeed, we conceived of the structure of the database with this goal in mind.

Navigocorpus therefore deals with a set of disparate sources, each of which provides data that ought to be stored with the greatest possible flexibility so that the information remains as close as possible to what appeared in the original source. At the same time, we wanted to make it possible to query the database without pre-determining the kind of questions or research goals of future users, who might be interested in very different aspects of maritime life, such as the items traded, the captain's life, rigs of vessels, duration of specific voyages or the evolution of shipping over time. Our challenge was to elaborate a structure that would be sufficiently flexible to deal with a wide variety of sources, an unlimited quantity of data which will progressively grow and a potentially infinite set of queries.

This essay describes the kind of technical problems we have dealt with and the solutions we have adopted. Further publications will show the considerable potential of the databases for historical research.

Navigocorpus has three main goals. The first is to store in a single database information on shipping drawn from a variety of sources, without any limit on the volume of data, in a way which makes the whole the data set a unique and homogeneously accessible universe. Second, we wanted to make it possible to add additional information about already documented events without breaking the existing links or requiring any changes to previously stored data. Finally, the project was designed to provide on demand the information subsets needed for specific research, in an immediately workable form, without requiring the user to make further changes to the data before processing them through other cartographic, statistical or analytical packages.

The core of Navigocorpus consists of a table of ship movements from one port to another.[3] A second table stores information about cargoes, while a third table contains information on duties, a less essential but nonetheless important type of information which is often provided in the sources, many of which are of fiscal origin. A fourth table holds data about various maritime actors (shipowners, brokers, cargo owners, shippers, consignees and the like). The system also provides assistance in locating and mapping ports and travels, de-

---

http://navigocorpus.hypotheses.org/.

[2]http://www.slavevoyages.org.

[3]A table comprises a set of records, each of which is the product of the aggregation of the same set of fields.

tails about monetary and quantitative units, the nature of goods mentioned (and their English translation) and a description of the ship types involved.

In providing a description of the structure of the Navigocorpus database,[4] we will focus here mainly on the table of vessel movements both because of its centrality and because it illustrates the main technical innovations. Cargoes and taxes will be treated more briefly given that their processing is fairly is based upon principles that are reasonably well known. Actors have been processed along a line which one of us has been developing over the past twenty years through the Fichoz system and which has already been described in various publications.[5] As to our approach to the description of goods, quantities and monetary units we shall say just enough to give the reader a rough idea of what we have done, in part because these parts of the project are still under development and in part because their complexity requires a separate paper.

## The Main Challenge: Loading Data about Ship Movements

Navigocorpus embraces all kinds of sources about shipping. It is based at present on documentation from ports, such as lists of entrances or clearances, drawn generally from registers about health, fiscal, consular and statistical controls. It focuses mainly on the eighteenth and nineteenth centuries, but the database is able to cope with information for any period.

In essence, Navigocopus is an open database and its structure reflects this. It has to be able to deal with information pertaining to the same voyage from various ports and sources. For example, a ship sailing from Genoa to Marseilles produces documentation in Marseilles which shows it entering from Genoa, while in the Genoese sources the same ship appears as clearing for Marseille. Both sources and the information they provide must be retained and made compatible. To put it in another way, the challenge consists of storing fairly complex data, drawn from a variety of overlapping sources, with the additional problem that there will never be a "closed set" of sources; additional data may be unearthed which will have to be added among partially overlapping existing sets without disturbing existing structures. A closer look at various possible cases will illustrate this point.

Take, for example, a simple form of data, such as an entry taken from registers compiled by the Health Office in Marseille:

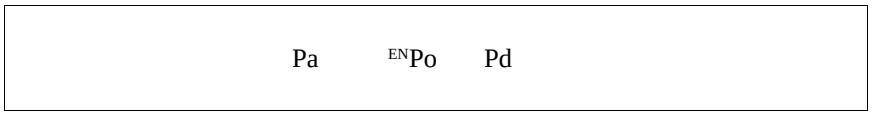
---

[4]Although we will not comment on the intrinsic content and quality of available sources, we recognize that such considerations are necessary before beginning research. In this paper, discussions of sources have been limited to their formal characteristics because these have a significant impact on the structure of the database. Nonetheless, we will also touch on some consequences of the model.

[5]Jean-Pierre Dedieu, "Les grandes bases de données: Une nouvelle approche de l'histoire sociale. Le système Fichoz," *Revista da Faculdade de Letras HISTÔRIA*, III, No. 5 (2005), 99-112.

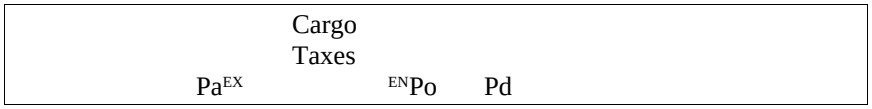
Example I: Marseille [Po] on 20 April 1787 [$^{En}$], entrance of the vessel *Orion*, 450 tx., captain Dubarry, from Smyrna [Pa], bound for Bordeaux [Pd].[6]

We call such an entry a "documentary unit." This means that all the information it contains comes from a single entry in a given source. The information provided by this documentary unit can be schematized as follows:

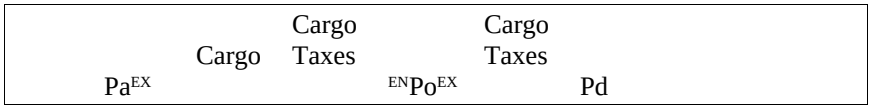



where Pa is the departure port, Pd the final destination, Po the port in which the movement has been observed (observation port), and $^{En}$ is the date the ship entered the related port.

Apart from the ports at which the ship called, the document may mention other elements, such as the cargo and duties paid when entering the observation port, and/or the date on which the ship sailed from the departure port ($^{Ex}$). This makes things a trifle more complex:



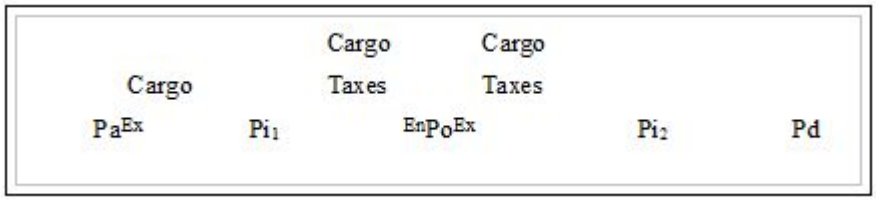

Additional information on the cargo at the port of departure; on the taxes and cargo when entering and leaving the observation port; and on the date of departure [$^{Ex}$] adds even more complexity:




The source might also mention intermediate ports [Pi] between the port of departure and the observation port [$Pi_1$], or before the final destination [$Pi_2$], and such information might or might not be associated with a date.[7]

[6]Elements in square brackets have been added to facilitate the understanding of the following diagrams.

[7]Dates are rarely expressed in an absolute way (e.g., 1 December 1747), except at the observation port. In most cases, they are provided as duration from the observation date ("arrived in thirteen days," "arrived in two months," etc.). We took this feature into account when structuring the database and added a routine to convert those durations into actual dates automatically. Dates in Navigocorpus, as well as in every database we write, are expressed as alphanumerical strings. We do not use the date formats provided by database packages because most of them do not manage relative dates (such as dates ex-




So far we have been considering cases in which all the information is derived from a single entry in one document. The one which served to build the structure immediately above would read like this:

Example II: Marseille [Po] on 20 April 1787 [En], entrance of the vessel *Orion*, loaded with dried figs and barley, captain Dubarry. Sailed from Smyrna [Pa] two months ago [Ex], loaded with dried figs, called at Messina [$Pi_1$]. Sailed on 27 April [Ex] to Barcelona [$Pi_2$], loaded with soap, bound for Bordeaux [Pd].

The same ship might be mentioned again, either in the same source or in another one, without any overlapping. This would be the case, for instance, if a ship returned to the same port at a different date.

Example III: Marseilles [Po] on 20 July 1787 [En], entrance of the vessel *Orion*, loaded with wine and spirits, captain Dubarry. Sailed from Málaga [Pa] two weeks ago [Ex]. Sailed on 29 July [Ex] to Genoa [$Pi_2$], loaded with salted fish, bound for Messina [Pd].

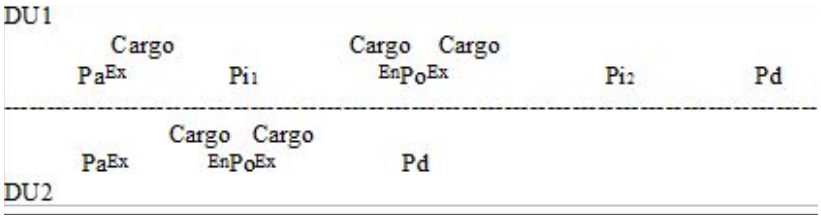
In this case, we just add the new documentary unit to the database. Thereafter, we will assign the same identification code to the ship name of these two documentary units to make it clear that it is the same ship at a different time (different ports, dates and cargo items). But in some instances two administrative units at the same port might produce two different records of the same event, each potentially telling part of the story:

Example IV: Messina [Po] on 10 April 1787 [En], entrance of the vessel *Orion*, loaded with dried figs, captain Dubarry. Sailed from Smyrna [Pa] forty-nine days ago [Ex], loaded with dried figs. Sailed from Messina on 15 April [Ex], loaded with dried figs and barley, bound for Marseille [Pd].

---

pressed as durations from another event) and because most are incompatible when shifting from one package to another. Within Navigocorpus, an absolute date is expressed in the following form: yyyy=mm=dd (year=month=date). Therefore, yyyy<mm<dd means a date preceding the expressed value, while yyyy>mm>dd is a date subsequent to the expressed value. Dates attached to a point are stored in two fields, one for the start of the dated event, another for its termination.

The example immediately above describes part of the travel mentioned in Example II. These two documentary units in the database provide two interlocking documentary entries, which we will respectively call DU1 (based on example II) and DU2 (based on example IV). $Pi_1$ of DU1 is the same point as Po of DU2.




While loading data from the first documentary unit, the researcher would of course have no idea that the second source exists. Similarly, in designing an open database, the creator will have no idea what interlocking documentary units will be found or where they will interlock. We therefore had to structure the movement table of Navigocorpus to be able to cope with such uncertainties.

Moreover, the number of ports mentioned by a given documentary unit varies widely: from our experiences with the sources we have exploited thus far, these can vary from one to twelve. The facts that we are dealing with potentially overlapping sources and ships which might have called at a large number of ports affected the structure of the database and determined the choice of the table we placed at its core.[8] Contrary to most existing databases on shipping, which deal either with a homogeneous, non-overlapping source or with a predefined, closed set of data, we could not set the documentary unit as the basic unit of the table's structured record.

We were firstly tempted to structure our database on stages, or segments of a given voyage. A segment is composed of two consecutive points. Had we adopted this option, after adding a mention of the documentary unit (DU1 or DU2) and an identifier for each point (A, B, C, D, E), the elements would generate the following records:

[8]If we had chosen to make a record for every documentary unit, we would have had to implement a field for every port. Apart from other considerations concerning the retrieval of data, the fact that the number of ports mentioned in a documentary unit is potentially unlimited makes it impossible to create ahead of time a limited series of fields to give a true account of them. A common solution is to create n-fields for n-ports and to exclude "excess ports" (from n+1 onward) from the database. Many "scientific" databases are constructed in this way or consign to a "Remarks" field everything which does not fit into the structure of the database, thus making it virtually impossible to exploit the information efficiently. This type of design, however, contradicts the most basic requirement of database creation and introduces an unacceptable bias into the information by omitting key pieces of data.

| | | | | |
|---|---|---|---|---|
| ~~DU1~~ | ~~$Pa^{Ex}$~~ | ~~A~~ | ~~$Pi_1$~~ | ~~B~~ |
| DU 2 | $Pa^{Ex}$ | A | $^{En}Po$ | B |
| ~~DU1~~ | ~~$Pi_1$~~ | ~~B~~ | ~~$^{En}Po$~~ | ~~C~~ |
| DU2 | $^{[En]}Po^{Ex}$ | B | Pd | C |
| DU1 | $Po^{Ex}$ | C | $Pi_2$ | D |
| DU1 | $Pi_2$ | D | Pd | E |

where A = Smyrna, B=Messina, C=Marseille, D=Barcelona and E=Bordeaux.

The first and second lines provide information on the same leg of the ship journey, just as do the third and fourth lines. It would thus be possible to reconstruct the actual voyage and to avoid double counting by erasing these superfluous lines[9] after copying the third line's entrance date to the fourth line. The result would be as follows:

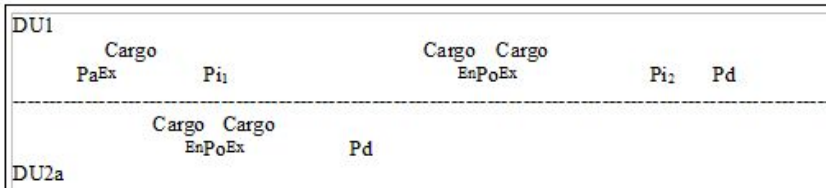
| | | | | |
|---|---|---|---|---|
| DU 2 | $Pa^{Ex}$ | A | $^{En}Po$ | B |
| DU2 | $^{[En]}Po^{Ex}$ | B | Pd | C |
| DU1 | $Po^{Ex}$ | C | $Pi_2$ | D |
| DU1 | $Pi_2$ | D | Pd | E |

We call this trimmed series of geographic segments, chronologically ordered and sailed by the same ship, a "route."

The scheme works as long as the second documentary unit displays ports already mentioned in the first one. If not, things become more complicated. Let us imagine a new documentary unit, DU2a, providing information on a journey we already mentioned in example IV (DU1):

Example V: Messina [Po] on 10 April 1787 [$^{En}$], entrance of the vessel *Orion*, loaded with dried figs, captain Dubarry. Sailed on 15 April [$^{Ex}$], loaded with dried figs and barley, bound for Nice [Pd].

DU2a names an additional port, Nice. The system must now be formalized in this way:




[9]We do not mean that we were factually thinking of expunging data from the database; instead, we mark them with a special identifier which allows the user to set them apart and not take them into account when calculating or retrieving data.

By structuring the database on segments, we would obtain:

| *DU* | *First port* | *First port id* | *Second port* | *Second port id* |
|---|---|---|---|---|
| DU1 | $Pa^{Ex}$ | A | $Pi_1$ | B |
| DU1 | $Pi_1$ | B | $^{En}Po$ | C |
| DU2a | $^{En}Po^{Ex}$ | B | Pd | F |
| DU1 | $Po^{Ex}$ | C | $Pi_2$ | D |
| DU1 | $Pi_2$ | D | Pd | E |

where A = Smyrna, B=Messina, C=Marseille, D=Barcelona, E=Bordeaux and F=Nice.

But something has gone wrong. We no longer have consecutive or repeated segments. To restore some degree of coherence, we must create a new stage, F/C (fourth line below) and copy and paste the entrance date in port C from DU1. No source explicitly mentions it, and for that reason we cannot provide it with a documentary unit identifier.

| *DU* | *First port* | *First port id* | *Second port* | *Second port id* |
|---|---|---|---|---|
| DU1 | $Pa^{Ex}$ | A | $Pi_1$ | B |
| ~~DU1~~ | ~~$Pi_1$~~ | ~~B~~ | ~~$^{En}Po$~~ | ~~C~~ |
| DU2a | $^{En}Po^{Ex}$ | B | Pd | F |
| [--- | Pd | F | [En]Po | C] |
| DU1 | $Po^{Ex}$ | C | $Pi_2$ | D |
| DU1 | $Pi_2$ | D | Pd | E |

where A=Smyrna, B=Messina, C=Marseille, D=Barcelona, E= Bordeaux and F=Nice.

Once this has been done, you just need to hope that the assumption that both documentary units refer to the same ship and voyage is correct. If not, when the mistake is apparent it is necessary to undo what you did and start again from the beginning. Such a cumbersome process may be acceptable if there are only a limited number of cases, a unique source and no need to add any additional data. But if the aim is to create a huge, flexible and open data-base, this solution is not viable.

We therefore decided to structure our database on the point, rather than on a segment of a journey. This choice, moreover, provides a much greater flexibility also in dealing with specific events, such for instance as the prize of a ship on open sea and the consequent change of captain.[10] Once the database is structured upon geographical points, it is much easier to eliminate redundant in-

[10]A British ship arriving at Marseille as a French prize during wartime, for instance, can be recorded as sailing from Smyrna (port of departure) bound for London (intended port of destination) with an English captain, but captured at an intermediate point ("twenty miles off Cape Passero") where the original captain is replaced by a French prize captain.

formation. The routes are easily reconstructed by setting points one after the other in the chronological order. The double information provided by points appearing twice is easily earmarked and set aside for exploitation purpose. We obtain thus a series of chronologically ordered points, instead of a list of segments. Examples II and V can be formalized as follows:

| *DU* | *Port* | *Port id* | *Port name* | *Rank in DU* | *Date entrance* | *Date exit* |
|---|---|---|---|---|---|---|
| DU1 | $Pa^{Ex}$ | A | Smyrna | 1 | | 20 Feb. |
| ~~DU1~~ | ~~$Pi_1$~~ | ~~B~~ | ~~Messina~~ | ~~2~~ | ~~>20 Feb.~~ | |
| DU2a | $En Po^{Ex}$ | B | Messina | 1 | 10 April | 15 April |
| DU2a | Pd | F | Nice | 2 | >15 April | |
| DU1 | $En Po^{Ex}$ | C | Marseilles | 3 | 20 April | 27 April |
| DU1 | $Pi_2$ | D | Barcelona | 4 | >27 April | |
| DU1 | Pd | E | Bordeaux | 5 | >28 April | |

In the figure above, the second line is redundant with the third one. Once you have eliminated the less precise information (line 2), you obtain a coherent route, ordered on the basis of dates, independently of the documentary units, which can now be mixed without inconvenience.[11] The date, a solid fact which transcends the relativity of the source, makes it possible to sort the points independently of the document from which they come. Practice over thousands of cases showed that this strategy is highly efficient and allows the interweaving of data from various sources with a high degree of flexibility.

Thus, we load data from different sources as they come. We provide every record – that is, every point mentioned by an entry in the sources – with a documentary unit identifier and a rank within the documentary unit (rank 1 for the most remote point touched by the ship during its journey, rank 2 for the following, etc.). Even more important, we attribute a date to each point; if no absolute date and no duration are provided in the source, we add a relative date (relative to the previous or the next dated event).[12]

Once various thousands of cases have been loaded into the Point table, we sort records by ship name and dates. Data are visually examined. The name of the ship, its burthen, homeport and the name of the captain are usually provided in the document and help to identify the ship. Once we believe that we have information on the same ship, we attribute an identifier to the vessel[13] and mark each point of its journey so that duplicated entries of the same ship at the same point and at the same date on its route can be set beside it without erasing the actual information provided by each documentary unit. In a few cases, a date

[11] We added a day to the Bordeaux date [$Po^{Ex\ +1}$] to make sure that the computer places this point after Barcelona, regardless of the context.

[12] See footnote 7.

[13] If there is any doubt, we assign two different identification codes because it is easier to aggregate than to disaggregate data.

of entrance or exit must be manually copied from one record to another when information is hopelessly split among various entries.

This process is time-consuming: a trained operator can process approximately 1000 new entries a day, and the identification of ships is tricky enough to be reserved to professional researchers only.[14] Yet this way of proceeding is the most efficient that we were able to conceive. This step can be easily repeated at any time after loading a set of new information to the database, for instance after having collected evidence from new sources. It is thus possible to expand indefinitely the route sailed by the ship over time.

A series of points proves much easier and much more flexible to manage than a series of stages. To add a new element, you simply need to add a place name instead of verifying the compatibility with previous existing segments of the route. This is why Navigcoorpus is structured on the Point table.

### Accuracy Markers

Sources tell different stories, and it is important when collecting data and putting them into a single database not to lose information on the kind of story they tell. Some sources refer to events which are situated in the past relatively from the moment in which the observation takes place at the observation point (the past of the past, to put it in Reinhart Koselleck's words).[15] This is the case of Málaga in example III, which is observed in Marseille as a past event. Some sources, on the contrary, provide information on intentions and future events, such as Messina in the same example, which is a future destination when observed in Marseille (a future of the past for us). Sources might also tell stories of a route which differed from the intended one, such as the case of a vessel which was wrecked or forced to enter a port because of a major leak, but which was actually bound for another destination it never reached. Most existing databases store this information, if at all, in a "Remarks" field where it is difficult to exploit.

We structured the Navigocorpus database so that we would not lose any information in the sources and so that we would be able to exploit such information efficiently and to analyze, for instance, journeys which never happened, such as the intended destinations of ships captured by privateers. In

---

[14]

The identification of the ship is in fact the most time-consuming task in the process. We do not believe that it is possible to make this step automatic, even when excellent contextual information, such as the ship name, captain's name, burthen and homeport, is available. This is not only because these names are often spelled in various ways in the sources but also because the problem is frequently exacerbated by erroneous transcriptions in the data-collection process. For these reasons, we have found that that no routinized programme works. The identification of vessels is so central to most types of analyses that we felt that was absolutely necessary to maintain, whatever the cost, the maximum quality on this issue.

[15]Reinhart Koselleck, *Futures of the Past: On the Semantics of Historical Times* (New York, 2004).

order to do these things, we have added information regarding the potentiality of every geographical point mentioned in the documentary unit ("past" or "future," with regard to the point and the moment when the story is told). We call this added information the "accuracy marker" and store it in a field called Point status." The list of markers added to each point when inserting the data into the database is as follows:

.P: past events
PC: past event declared as positive ("coming from Smyrna").
PA: past event declared as an alternative. Various possible starting points (e.g., "from Bergen or Oslo").
PG: past event declared as sailing around a gravity point (e.g., "fishing around"). The documentary unit in this case consists of a unique point.
PM: cumulative past event. Various starting points recorded without indicating their relative rank (e.g., "from Bergen and Oslo" when the chronological order of this points is questionable).
PS: static past event (e.g., ships reported as anchored in a harbour at a given date).
PU: intentional past event recording an unrealized past intention (e.g., record in Copenhagen of a Riga-Copenhagen journey in which a call at Danzig was planned but not realized; Danzig is marked PU).
.F: future planned events
FC: planned event declared as a positive intention (e.g., "clearing to Málaga")
FA: alternative planned events. Various destinations declared as equally possible, although mutually exclusive (e.g., "to Bergen or Oslo").
FM: cumulative planned events. Various planned destination declared as intended, without indicating their relative rank (e.g., to Bergen and Oslo).
FG: planned event declared as sailing around a gravity point (e.g., fishing around). The documentary unit is composed of one point only.

The addition of the accuracy marker consumes a little time when entering the data but is indispensable because it process allows a researcher, for instance, to work exclusively on past events or on false declarations, which may be quite common under certain circumstances, such as warfare.[16]

**Encounters**

[16]The Point status field is comprised not only of the accuracy marker but also of two alphabetical character markers which indicate the kind of sources from which the data have been extracted. This element has no structural role in the database, but it provides the user with useful information to interpret the data. A provisional list of the same is as follows: CO: certificate of origin; ML: marine list, list of ship movements; RC: consular register; RF: tax register; RS: health register; and TT: indirect information provided by a witness, without direct observation by the creator of the source.

Sources often report the meeting or sighting of a another ship. In this case, we create a new point in the correspondent documentary unit of the route of the two (or more) ships in question. This point is described as an "encounter" in a specific field. All ships which met or saw each other at a given point are provided with the same encounter identifier in order to allow the reconstruction of the overall encounters on each of the routes concerned.

**Technical Fields and Tables**

Navigocorpus is structured upon the Point table (see appendix I). Every point is provided with a record identifier comprising eight digits (00000000). This identifier is automatically set by the system every time a new record is created. It links the point to complementary data which depend on it, such as cargo items, taxes or actors' actions. Each record is also equipped with a documentary unit identifier which allows the user to retrieve all the points (i.e., records) which constitute a single documentary unit. Concretely, example 2 will appear as follows:

| **Date Entrance** | **Point Name** | **Date Clearance** | **Record ID** | **Point Status** | **Documentary Unit ID** | **Point Rank** |
|---|---|---|---|---|---|---|
| 1787>04>28 | Bordeaux | | 00000005 | FC-RS | 00000001 | 5 |
| 1787>04>27 | Barcelona | 1787>04>27 | 00000004 | FC-RS | 00000001 | 4 |
| 1787=04=20 | Marseille | 1787=04=27 | 00000003 | PC-RS | 00000001 | 3 |
| 1787>02>20 | Messina | 1787<04<20 | 00000002 | PC-RS | 00000001 | 2 |
| | Smyrna | 1787=02=20 | 00000001 | PC-RS | 00000001 | 1 |

Navigocorpus makes it possible to display on the same screen all the information provided by a documentary unit, a feature which makes data loading fairly easy, despite the fact that the database is structured on the points and not on the ship or captain's name (see appendix II).

**Linked Tables**

We have so far explained the Point table, which is main table of Navigocorpus and was by far the most problematic to conceive given that existing databases are generally structured in other ways. We will now briefly present the other tables that are linked to the main one.

*Cargoes*

Sources often describe cargoes partially or fully at various points along the route. Information on the cargo is thus stored in a specific table which is linked

to the relevant point. Each item of the cargo constitutes a record in the Cargo table. Each record is composed of a description block describing the commodity of which the item is composed (full text description), a quantity block describing the number, weight or volume of the commodity (quantity, unit) and a price block, describing its value.[17] A standardization field provides the English translation of the commodity name. The translation relies on a dictionary of commodities, which forms a separate file and has been independently elaborated as a specific part of the database. Navigocorpus does not cluster the items according to an imposed classification system and leaves the user free to do so according to his or her specific research issues. A specific coding field allows the user to group the items into as many and as specific classes as he or she wishes (e.g., "colonial goods," "fish", "wood") based either on the names of commodities or on any other available criteria (for instance, conditioning: which liquids are shipped in bottles at a given time?).

*Taxes*

Sources might mention taxes or duties paid at one point. These taxes are registered into a Taxes table. Each tax item forms a specific record, which is linked to the point at which the source mentions it was paid, and eventually to the cargo item on which the duty was levied. The record is composed of two blocks: a full text description of the tax (its name) and its monetary value.

*Actors*

Even if this is not their primary function, documents describing ship movements, such as port registers, health registers, fiscal registers and shipping lists frequently mention a number of actors in addition to the captain, which we considered as an identity marker of the vessel and insert together with the other information referring to the ship. Shipowners, consignees and supercargoes, among others, are frequently named.

The Actions table is the central piece of Fichoz, a database conceived to store all kind of social history data which Jean-Pierre Dedieu developed over the past two decades. Generally speaking, a new record is added to the Actions table every time an actor performs an action. In Navigocorpus, this record is linked to the point of the Points table to which the information on the action is related. An action is defined by an actor's name (Sir Cosmo Parkinson), a description (e.g., Secretary of States for the Colonies), a date (1940, May 13), a

[17]Given that monetary values are expressed either in a bipartite (unit/cents; unit/*millimes*) or, in early modern accounts, a tripartite system (pounds, shillings, pence), Navigocorpus (like every other database we created) stores them in three pairs of fields. Each pair describes either the first and second levels of a bipartite system, or the first, second and third levels of a tripartite system. The first leg of the pair contains the quantity; the second one the unit. For example: 67/French francs//34/*centimes*//-/ - or 4/sterling pounds//13/shillings//5/pence.

place (London) put in a constable's textbook form – in other words, an action is characterized by the following questions: who, what, when and where. Fichoz, and consequently Navigocorpus, adds a fifth criterion, "with whom" (e.g., a minister in Winston Churchill's government). Every time the answer to one of these questions changes, a new record is created. If John Smith is both the shipowner and the cargo owner of the ship *Orion*, this produces two entries in the Actions table because the action is different, even if all the other elements are the same.

In this way we are able to store an unlimited number of informational items about an unlimited number of actors – individuals as well as firms or corporations, such a shipping companies. It should be stressed that this information might be mentioned in relation to a ship route or independently in other sources (all sorts of legal documents, such as wills, contracts, legal settlements, lawsuits, accounts, etc.; insurance contracts; chronicles; business correspondence, etc.). This information interweaves with the entire set of information provided by specific maritime documents. Fichoz provides a table to store textual or iconographic documents and to link them electronically to every action connected by a substantial conceptual link. Fichoz is also able to process genealogical links between actors.

The notion of an actor in this context is quite broad. We consider an actor to be any entity which serves to support relationships with other entities. In this sense, people, corporations or even objects may be an actor: a paper contract is an intermediary between actors.[18] We can even consider a ship as an actor; this makes it possible to write its life story, from building to scrapping and all the intermediate events, as a list of actions, and thus to treat every crossing of any geographic point as a specific event. This in tuen enables the user to process a relationship with any actor as a relationship between actors. In other words, the Actions table makes it conceivable to insert Navigocorpus events and ship movements into the global flow of social history in the broader meaning of the concept.[19]

### Auxiliary Tables

Navigocorpus provides a series of auxiliary tables which facilitate the understanding of its contents and the exploitation of the information it contains.

#### *Sources*

Navigocorpus shares with Fichoz a Source table which contains a description of all bibliographical references and archival units used as sources. This table is

[18]See Bruno Latour, *Reassembling the Social: An Introduction to Actor-Network-Theory* (Oxford, 2005) for hints about the underlying sociological theory.

[19]See Dedieu, "Les Grandes bases."

linked with a "Source" field which appears in all other tables. A special routine, based on this link, can call up on demand a summary of the source from the Source table.

*Diem*

Navigocorpus also shares with Fichoz a *Diem* [*Dictionnaire institutionnel de l'Europe moderne*, or Dictionary of Institutions of Early Modern Europe] file. *Diem* contains a description of historical concepts and institutions. Users are welcome to provide their observations and the conclusions they reached to progressively expand the *Diem* table.

*Help*

The Help table contains detailed information about the concepts on which Navigocorpus and Fichoz have been structured; descriptions of every table, file and field in the database; descriptions of each programmed routine with which the system has been equipped; depictions of every screen layout, with sketches; instructions on how to fill in data and cope with difficult cases; and explanations on how to retrieve data and export them to downstream processing packages. The Help table is available in English.

The Help file is autonomous; it may be opened and searched through queries on its own. Records are physically linked to each other based on their conceptual proximity and can be accessed "in chain." The Help table can also be accessed from any other table through dark green triggers. Larger ones open the Help file on the last selected record, while smaller ones send information to the description page of the selected screen layout of the table from which Help has been requested.

*Geogeneral*

Geogeneral is a geographic gazetteer. At present, it contains two million georeferenced points and encompasses Europe, the Mediterranean coasts and the Americas. We intend to expand it to the rest of the world in the future. Each point is associated with an open set of names, which allows queries based on any possible spelling (Nueva York/New York) or historical denomination (Istanbul/Byzantium/Constantinople). Each point is also provided with a specific "UHGS identifier," used as a basis for its integration into maps and other geographic processing packages. Once data have been extracted from Navigocorpus according to a given set of criteria chosen by the user, it is possible to create a map with the specific software within a couple of minutes.

**Data Retrieval Strategy**

To store data is one thing; to retrieve them from the warehouse in which they have been stored and to process them to come to sound conclusions is another matter. We consider that a database is intended to store data in the most neutral way possible to allow a researcher to process them in the way he or she considers most suitable. Generating information implies the introduction of some kind of cognitive structure. Another structural injection is necessary to transform information into data, with a capacity to be stored, retrieved and used again when necessary. Without these cognitive structures, the human mind could not control any material, and storage would be impossible. The sources we use to write history are, in this sense, warehouses of previously structured data.

Navigocorpus, like any other scientific database, adds some extra structuring tools which make a broader and more efficient processing of the information possible. It provides, for example, a kind a pre-treatment of the information delivered by the source. This must meet contradictory needs. For one part, it must be powerful enough to generate a significant improvement of the source, a requirement which involve ssome kind of high-level structuring. For another, it must be broad enough to meet the needs of the largest possible esearch community, a requirement which tends towards low-evel structuring. Navigocorpus provides a structure centred on the most essential aspect of shipping: objects (ships and the goods and people carried on them) moving through a continuous and chronologically ordered series of points. Our database is thus founded on the basic organizing concept of any kind of information about shipping; this made it possible to collect into the same "container" and to transform the information delivered by a huge variety of sources into data along exactly the same lines. Even more, this choice makes it possible to connect to shipping data much extra information about goods, actors, etc., which are just as dependent on time and space. We thus provide the researcher with means of accessing freely a huge territory which, some time ago, would have taken a researcher an entire lifetime to explore even in a very partial way.

Using concepts as basic as time and place to disaggregate information into data not only offers efficiency but also a broad scope of potential utility. We carefully refrained from introducing any higher-level structuring concepts. We identify times and places in the most pedestrian way, as a name, location within a time span and a set of coordinates. We do not even describe a point by the fact it belongs to a specific country or continent. In other words, low- level structuring criteria enhance the basic articulations of the data provided by the sources and make them immediately available for any kind of research. But imposing only minimal and low-level articulations on the data makes it difficult to retrieve them in view of research strategies which demand a higher level of formalization, the creation and characterization of a variety of specific classes of objects and other structuring processes which recast the data based upon the user's needs, hypotheses and prejudices.

Navigocorpus provides a way of achieving such high-level formalization without changing anything in the original data. Every table is mirrored by a Dictionary table, a set of records the identifiers of which reproduce the record

identifiers of the original table and create a link between the original and the Dictionary records. The Dictionary record comprises a coding field in which users may characterize the corresponding data with as many characters as they wish. In such a way, every piece of stored information can be described with as many dimensions as users think proper.

Special routines have been embedded in the system to make this process easier. The user gets through two easy steps. The first consists in selecting within the database a subset of pertinent records that he or she wishes to characterize with the same code. The second step consists in activating the marking routine, which applies the code to all the selected entries. The code may consist of any string of characters, provided they form a unique word. For instance, a researcher interested in the fish trade, might use the coding field to add the word "fish" to all the documentary units where the cargo encompasses cod, herring, tuna, etc.

Dictionary tables are not treated as parts of the data files but as independent files. This makes it possible for each user to have his or her own Dictionary tables and to couple them on demand to the main data files just by changing the Dictionary file's name.

Finally, both the codes and the original data can be exported to other processing packages, usually a spread-sheet in tabulated form. Navigocorpus provide intermediary files to transform and pre-treat data, if necessary, to make them compatible with some special demand of specific downstream packages (e.g., Pajek, a network analysis package). If necessary, new derived characters brought out by the analysis can be added to the Dictionary and can form a basis for a new characterization of the original data.

## Technical Specifications

### *Database Package*

Navigocorpus has been developed under FileMaker®, a database package of Claris Corporation. Its basic tables and field structure are nevertheless independent of FileMaker and can be implemented easily under any relational database package. It is highly desirable, however, that this alternative package allow an easy and quick retrieval of any string of characters independently of its position within a given field and does not require left to right searches. Much of Navigocorpus' effectiveness in managing large volumes of information is based on FileMaker's efficiency in retrieving data with an impressive flexibility. The database has been globally designed to take this aspect for granted. A less efficient and versatile retrieving tool would probably require important structural changes in the basic architecture of the database to preserve its present efficiency.

### *Ergonomic Tools*

If working with database packages other than FileMaker®, it would be also highly desirable for the package to make it possible to generate easily any kind of screen layout on demand. Navigocorpus stresses ergonomy, which is a basic factor of efficiency. A special screen layout is provided for every operation. A long list of routines, called up by triggers displayed on the header of the layouts or scattered near the correspondent field, allow sorting, retrieving and a set of other usual processes to be performed through a simple click. A colour code is systematically used to characterize each part of the screen layouts through the database. Although Navigocorpus might appear complex at first sight, we believe it is actually as user-friendly as possible given the intrinsic complexity of dealing with information different in nature and possibly providing overlapping situations.

The on-line Navigocorpus database will offer an even more user-friendly environment for queries only, but will not contain all the potentialities offered by the database we created. We intend, however, to promote the broader use of Navigocorpus and enable researchers to access the database directly on demand without passing through the on-line interface.

## Conclusion

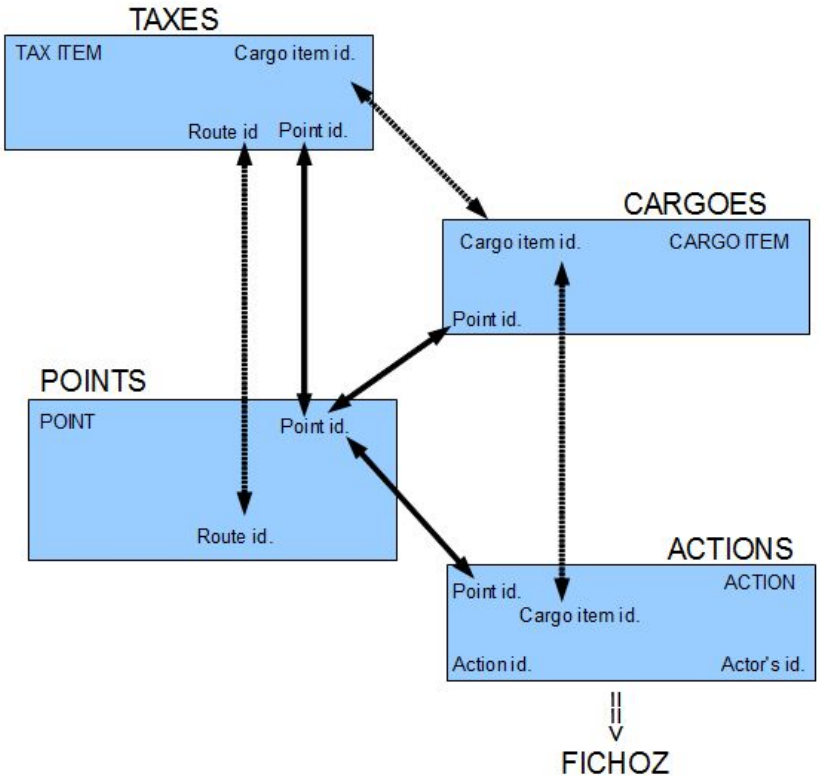
The four-year ANR programme which enabled the creation of Navigocorpus was conceived as a methodological challenge. We wanted to build a database which would be able to store and process all sorts of information on ships and their movements provided based on all possible sources. We believe that we have acomplished this mission.

Our most important contribution is probably the concept of the "point" as the basic element for structuring a shipping database. The term "point," as we use it, does not simply mean a geographical location. It is that, of course, and as such it must be described by a set of spatial coordinates (longitude, latitude, altitude, position on a specific spheroid, etc.). But this geographic location is also the set of an action, a place where an action takes place. We call a point an entity characterized at the same time by a location, a time set and an action set performed by specific actors. All these characters taken together (place, time, actors, action) define the point as we manage it in Navigocorpus. When one of those elements changes, another point is created: the entrance into Toulon harbour of the vessel *La Licorne* is one point; the entrance at the same time into the same harbour of the frigate *La Sardine* is another.

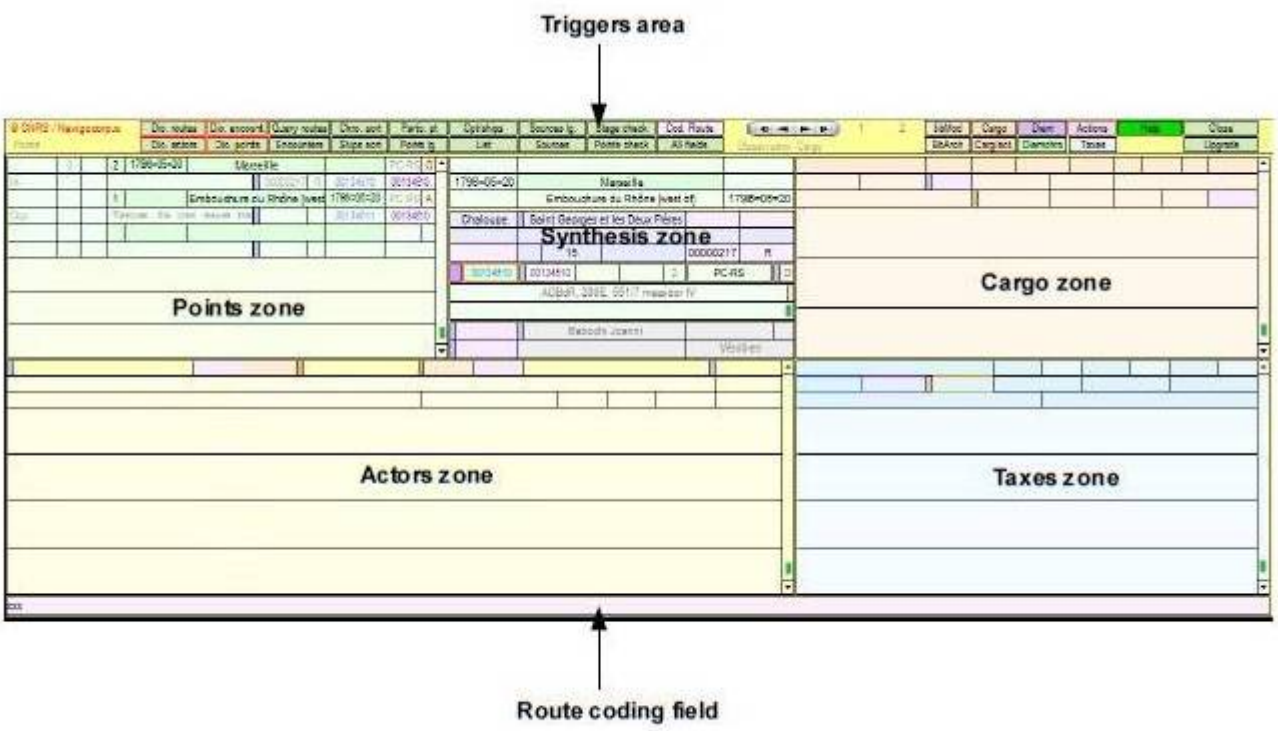
Organizing data in this way is the key to Navigocorpus. It works surprisingly well, and we hope that the scholarly community will share this assessment. We believe this was the only choice to enable the merger of other databases, provided they have been soundly structured. Although Navigocorpus currently contains some tens of thousands of ships, it looks like a mere drop in an ocean of sources. The future will tell us whether this drop will ever become a

wave on which it will be pleasant to surf to observe phenomena noone could previously see.

**Appendix I**
**Navigocorpus Table Structure**



**Appendix II**
**Navigocorpus Main Layout**



**Appendix III: List of Files**

While it would have been possible to merge Navigocorpus and Fichoz tables into one file, once the tables were merged it would have been difficult to separate them. Since we believe in modularity, we decided to leave each separate and to link them where necessary. We thus can easily create specific sets of tables as required.

*a) Navigocorpus*

1) Data files

Navigocorpus_cargo
Navigocorpus_points
Navigocorpus_taxes

2) Dictionaries

Navigocorpus_dictionary_commodities
Navigocorpus_dictionnary_encounters
Navigocorpus_dictionary_ items_cargo
Navigocopus_ dictionary_pointdocs
Navigocorpus_dictionary_points
Navigocorpus_dictionary_routes
Navigocorpus_dictionary_tax_sequences

3) Auxiliary files

Fichoz_help: Navigocorpus and Fichoz help file.
Geogeneral: a list of identified and geo-referenced geographical points.
Navigocorpus_measures: a dictionary of measures

*b) Fichoz*

1) Data files

Actoz_actions
Actoz_arrays_D2: statistical data in two-dimensional arrays
Actoz_documents: content of the documents actions have been extracted.
Actoz_genealogy: a special file to process genealogical links between actors
Actoz_geography: a special file to process data from corographies and geographical dictionaries
Actoz_sources

2) Dictionaries

Actoz_dictionary_actions
Actoz_dictionary_documents
Actoz_dictionary_genealogy
Actoz_dictionary_geography

3) Auxiliary files

Actoz_chronology: Noteworthy events used by the sources as date for the data.
Actoz_diem: a dictionary of institutions and historical concepts