Supply-side historical information systems: the use of historical databases in a public record office
Boonstra, Onno W. A.

Veröffentlichungsversion / Published Version
Zeitschriftenartikel / journal article

Supply - Side Historical Information Systems
The Use of Historical Databases
in a Public Record Office [1]

Onno W. A. Boonstra*

1. Four major problems in developing historical information systems in record offices

Most computer-projects in the record office are in a stage of experimentation. Some projects have been set-up to computerize the office administration, others deal with computerized inventories of archives. There are also projects which are designed to store the original contents of archives into a database system. The latter ones are without any doubt the most audacious and, at least for historians, the most interesting.

Their number, however, is comparatively small. There are four reasons why. First of all, a project of this kind takes a long time to get completed. In an experimental stage, short-term projects are favored. Secondly, historical data cannot be put into a database that easily. In order to meet the requirements of an information system, it is preferable to use data which are correct, consistent, complete and clear. Historical data rarely can be labelled as such (Boonstra 1986). Thirdly, the construction of historical information systems is hampered by the presence of multiple, sometimes conflicting preconditions set by the system's input and output. The output of an information system depends on the questions one wants to solve; most information systems, therefore, are developed according to the demand-side of the system. When constructing a historical information system, the supply-side of the system is fixed by definition and cannot be disregarded. The last problem in building a historical information system has to do with the way original sources are put into the system. A historian will select from a source only those data which are relevant to his research. A record office will demand that data are transferred into the system without any mutilation (Sieglerschmidt, 1988).

---

* Address all communications to Onno W. A. Boonstra, Fakgroep f. Geschiedenis, Katholieke Universiteit Nijmegen, Postbus 9103, NL-6500 HD Nijmegen, The Netherlands.

2. A supply-side information system

In this article, I will discuss the possibilities to solve the problems mentioned above and to create a source-oriented, supply-side historical information system. A particular system, which eventually was labelled »Magic Nana« 2), will serve as an example. It was commissioned by the Streekarchief Zuid-Oost Brabant and developed during the 1987 under-graduate course »Archiverings-systemen« at the History Department of the University of Nijmegen. The system is based on the 1850 and 1860 population registers of the community of Borkel en Schaft, a small village in the province Noord-Brabant.

The Public Record Office of Zuid-Oost Brabant stated that the system should meet the following requirements:

- all data must be transferred into the system
- data must be kept in its original form, i.e. without any pre-coding
- it must be easy to retrieve information from the system
- the system must give information about all levels of data that is in the registers, i.e. about persons and houses
- the system must be static as well as dynamic, i.e. it must be possible to retrieve information which is correct at a given date and retrieve information which spans a certain period of time.

These requirements concern both, input and output, i.e. the supply-side and the demand-side of the system. The supply-side prerequisites, howe-ver, are much more strict and were, therefore, considered the most impor-tant ones.

3. Problems with Population Registers

A population register is a very valuable source because in it a number of personal events are described which happened to the inhabitants of the community during the ten-year-period between two censuses. Informa-tion is given about birth, place of birth, marital status, death and migra-tion, about the occupation of the residents and their religion. With these data one is able to reconstruct a person's history and the life-cycle of a household. Because a population register is structured like a data-matrix, it does not seem too difficult to put the data into a database. There are, however, some problems:

1. There are different types of registers with different sets of informa-tion. In our case, the 1850 register differs from the 1860 register.
2. A considerable number of categories does not contain data. In some cases this is because information is really missing, in some other cases data have not been recorded because it was found too much trouble to

state superfluous information.

3. Some categories contain double data. Information has been erased and replaced by some other information.
4. The spelling of identical information sometimes changes. This holds especially for the spelling of people's names.
5. People can appear in a register more than once.
6. Sometimes categories contain information which in fact belong to another category.

4. **Magic Nana: filestructure**

In a population register, three levels of information can be discerned: the register as a whole gives information about the source, every page in the register gives information about a house, every line gives information about a person. A division of these levels of information into three separate files, therefore, seems appropriate. Relations between the files can be made with codes which are appointed automatically: the source-code is put in the source-file as well as in the house-file; the house-file is put in the house-file as well as in the persons-file.

Such a division of information into three different files gives no solution to the problems mentioned above. These problems, however, have been solved by extending the number of datafiles and by creating some special computer programs. The problem of differences of information between registers may very well lead to a number of space-consuming »empty cells«. It is preferable to put those categories« which are not present in all sources into separate files. Data which are present in exceptional cases only and which are, therefore, structurally missing most of the time, can also be put into separate files. For instance, next to the person-file, there will be a »migration»-file and a »decease»-file, too. Data which are missing but should be present are put into the files as well. The problem of double data can be solved by putting the data into separate records. About the differences in spelling: our aim is to put uncoded, original data into the system. In order to link data that have the same meaning but are spelled differently, a standardization program enables users to define their own kind of standardi by creating a »standardized« file which serves as a thesaurus of original and standardized data. The difference in spelling of person's names has been solved by creating a meta-person-code. A person which is present in the sources more than once, possesses also more than one person-code. In order to link all data of that particular person, he must be awarded with a »meta-person-code«. For houses the same holds true. A special linkage program had been made to make this linkage possible. If it is clear that a category is filled with data that belong to *another* category, these data are put into the right category.

In this way, a system is created which consists of dozens of files. This is not a big problem: a modern relational database management system is able to deal with a unlimited number of related files concurrently. Unfortunately, in our course we were forced to work with dBase IlI + , a system with which only a maximum of 15 files can be consulted at the same time. It was, therefore, decided to condense the system into six files only: first of all, there is a person-file which consists of all categories that are nearly always present in the sources. This file is called the PERSSTAT-file. Next to that, there is a PERSDYNA-file which consists of data that are present either in only one type of register, or only sometimes, or more than once. Data about houses are split the same way. There is a HOUSSTAT-file which consists of categories that are invariable, such as the page-number of the source and there is a HOUSDYNA-file in which variable data (address for instance) are put. Data about sources are put into a SOURSTAT-file.

The STANDARD-file contains the thesaurus of original and standardized data.

PERSSTAT = (<u>perscode</u>, meta-perscode, housecode, date of registration, name, christian name, date of birth, place of birth, sex, religion)

PERSDYNA = (<u>perscode</u>, <u>type of data</u>, <u>date</u>, description)

HOUSSTAT = (<u>housecode,</u> meta-housecode, sourcecode, pa gen umber)

HOUSDYNA = (<u>housecode</u>, <u>type of data</u>, <u>date</u>, description)

SOURSTAT = (<u>sourcecode</u>, type of source, community, date, screen-definition)

STANDARD = (<u>original datum</u>, <u>type of data</u>, standardized datum)

5. Magic Nana: software-program

The system consists of five different programs: one with which data can be entered, one with which data can be corrected, one to link records, one to standardize data and one to retrieve information.

Data-entry

A data-entry program is meant to put data into the system in such a way that the speed of entry is maximized and the chance of making errors is minimized at the same time. In order to do so, the Magic Nana data-entry program adjusts itself to the original sources. After some questions on the source's type, community and date, the system looks into the SOURSTAT file to look for the corresponding screen-definition number. According to this number, a screen is displayed which depicts the lay-out of a line in the register.

Data can then be entered; at the end of a line the system asks if there

still are data in the source that have not been accounted for. If so, the user gets the opportunity to enter these data as well. In this way, it is possible to deal with categories which have been filled twice. When all data are entered, it is into the relevant files.

Correction

This program is meant to correct errors after data have been entered.

Linkage

When the stage of data-entry is completed, it is necessary to check whether a person or a house appears more than once as a record. If this is the case, both records should be appointed the same meta-person-code or meta-house-code. The linkage-program is meant to assist in deciding whether a link should be made. By controlling for similarities in names, dates of birth and places of birth, two records are displayed which are more or less alike. The user can follow the suggestion: then a link will be made.

Standardization

This program is meant to build up the STANDARD file. It is possible to make standardizations on place, relation, marital status, religion and gender. The advantages of standardization over coding have already been stated: all original information stay into the system and will not get lost but it, nevertheless, is possible to use them in standardized form.

Retrieval

The retrieval program consists of a comprehensive set of modules which display all possible kinds of information onto the screen. The program starts with some questions: does the user want to see information in original or in standardized form? Does the user want to look for houses or for persons? If a house, what address? If a person, what name? Does the user want to see a history or only wants to see information which are correct at a given date? If so, what date? And so on.

6. An evaluation of a supply-side system

When a public record office settles for public accession to their historical data by way of an information system, it will first have to bestow care upon the supply-side of the system. All data must be put into the system; nothing can be left out; data must be entered without any kind of pre-coding. Only when these preconditions are met, it can pay attention to the demand-side of the system: it will have to be possible to give answer to the widest variety of questions.

In this article, attention was focused on the supply-side of a historical information system. It was shown that in spite of some serious problems

connected with historical data and historical sources, such a system can be made. Data which relate to different levels of information can be put into different files; data which are not susceptible to change can be put into a static file; data which do change over time can be put into a dynamic file; data which belong together, but are collected in separate records can be interrelated by means of »meta-coding«; whenever one wants to use standardi data instead of original data, it is possible to construct a thesaurus according to one's own specifications.

Magic Nana, the system we developed at Nijmegen, is a dBase III + application. Although it was not meant to become a professional application to be used in a public record office, its only purpose being to experience all problems connected with the construction of a supply-side information system; people who are interested in the system can ask for a copy.

Notes

1. Magic Nana, the information system to **be** discussed in this article, could be constructed only with help of F.E. Ector of the Public Record Office of Zuid-Oost Brabant in Eindhoven and the students of the »Archiveringsssytemen« course of 1987.
2. Magic Nana is a character from Jan Cremer (1966). Nana stands for »Nijmegen Automatiseringsproject voor Nederlandse Archieven«.

References

BOONSTRA, O.W.A. 1986. Over de betekenis van de informatica voor de geschiedwetenschap. Paper presented at the congress »Balans en Perspectief«, Utrecht, May, 1986.

CREMER, Jan. 1966. Ik Jan Cremer, II. Amsterdam, de Bezige Bij.

SIEGLERSCHMIDT, Jörn. 1988. »Probleme des Aufbaus und Umfangs einer Datenbank 'Historische Statistik'.« Historical Social Research/ Historische Sozialforschung 13, 89-110.