COMPUTER-ASSISTED LONGITUDINAL SURVEYS DESIGN AND IMPLEMENTATION

Assoc. Prof. Dr. Claudio CAPILUPPI•

Abstract

In the social research field, the longitudinal method is used when the study is about the evolution of a time dependent phenomenon. In longitudinal surveys, the same subjects are followed across time detecting changes on their measured variables, by means of a questionnaire that can be dependent on time and historical data of each subject. Time dependent interviewing is one of the main focus in longitudinal surveys, but it is very difficult to implement in practice by paper and pencil interviewing. The introduction of CAI (Computer-Assisted Interviewing) in longitudinal surveys promises to revolution this field of surveys, making actually possible to implement advanced interviewing techniques by using individual historical data. In this paper we discuss about the technological support required in order to implement longitudinal surveys with intensive time dependent interviewing, and suggest some solutions for a CAI system with full longitudinal support.

Keywords: computer-assisted interviewing, CAI, panel surveys, dependent interviewing, questionnaire, data model

• Department of Psychology and Cultural Anthropology, Faculty of Educational Sciences, University of Verona. E-mail: claudio.capiluppi@univr.it

1. Introduction

In longitudinal surveys, a panel of subjects is followed across time, detecting individual changes on measured variables, in order to understand the dynamics of some process. The same respondents are repeatedly contacted at time intervals, and administered by means of a questionnaire that can be different for each "wave" of the survey. In order to seize individual changes, longitudinal surveys are likely to have very complex questionnaires, with a large number of questions and complicated fill in "paths" defined by a number of routing rules, that can be extremely difficult to administer by paper and pencil interviewing, in particular when intensive time dependent interviewing would be implemented.

Dependent interviewing makes use of individual information available on subjects, coming from previous surveys or other data sources, in order to customize the current interview in different ways. Time dependent interviewing is used in longitudinal surveys for a lot of purposes (Jäckle, *et al.*, 2007; Lynn, Essex, Social, & Research, 2005), such as: to probe for changes, to avoid asking already known information, to give a feedback before asking new information, to remember to the respondent a previous answer, to examine more in depth some cases of particular interest, to customize the questionnaire to the wave of the survey and to the individual respondent, to check for inconsistent answers, etc. Dependent interviewing is one of the main focus in longitudinal surveys, but it is very difficult to implement in practice by classic paper and pencil interviewing (PAPI) because of intrinsic logistical difficulties and "hardware" constraints of the method (Banks & Laurie, 2000). Think only the need of customizing a printed paper questionnaire for virtually each individual case, or the need for the interviewer to access historical information about the current respondent, on related papers, in order to follow the designed question flow. It is needed an enormous work to prepare materials for PAPI interviewing at each wave of the survey, and for all these efforts, we have an high risk of errors in the questionnaire administering and filling in, producing non-pertinent answers and missing responses.

The introduction of computer assisted interviewing (CAI) in longitudinal surveys promises to revolution this field of surveys, making really possible to carry out longitudinal surveys implementing advanced interviewing techniques, using historical data on subjects, such as reactive probing techniques, proactive feedback techniques, consistency checks across times, etc. Computer assisted interviewing allows for handling the complexity of a longitudinal questionnaire, virtually without errors, because the questionnaire routing logic is administered by the programming logic built in a software system. By this way, CAI allows for implementing sophisticated dependent interviewing also in self administered interviewing (Brown, Hale, & Michaud, 1988; MP Couper & Nichols, 1998; Ramos, Sedivi, & Sweet, 1998). By means of automated question flow administering and real-time consistency checks, CAI provides better control on nonsampling errors, improving the survey data quality (De Leeuw, Hox, & Snijkers, 1995; Fuchs, Couper, & Se, 2000).

But, all that glitters is not gold… These great features, we have to make it clear, are quite theoretical, that is, an ideal CAI system could, in principle, implement in a proper way the techniques we are thinking to use in longitudinal surveys. Unfortunately, the currently available CAI systems have not been designed for longitudinal surveys and do not provide specific support for longitudinal time dependent interviewing, limiting what is actually possible to put into effect *versus* what we could think to do.

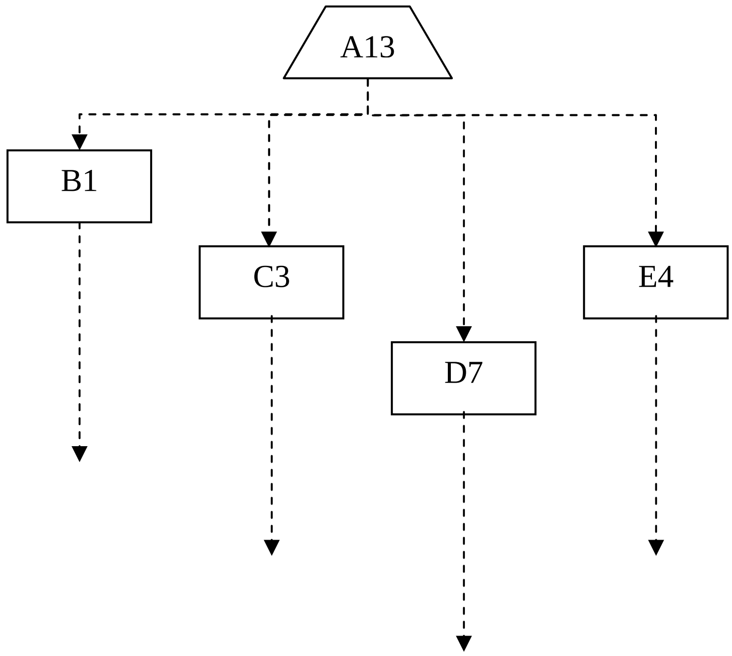
2. Computer assisted questionnaire design

The questionnaire is commonly seen as a sequence of questions with an administering logic defined by a number of rules (Bethlehem, 2000), consisting of conditions on subject responses that route the interview to the next question. Passing from the traditional PAPI to CAI, we have to introduce a new way of considering the questionnaire, namely as a *data model*. Essentially, a data model is a conceptual abstraction to represent some aspects of a realty of interest. According to the *relational database theory* (Atzeni & De Antonellis, 1993), a realty can be represented as a set of *entities* and *relationships* among them. We can term *conceptual modelling* of the questionnaire, the design process of a proper data model to represent the survey response data[1].

In conceptual modelling of a questionnaire, each *concept* (each question) can be represented as an *entity* or as an *attribute*. The modelling process is driven by a well defined methodology (Kent, 1983), where the key principle is avoiding *redundancy*, that is the main concern in data base applications. Redundancy is the consequence of a conceptual modelling error, in fact it derives from over-representation of a concept. For instance, we originate redundancy when, modelling a questionnaire where a same question appears in different sections, we represent each instance of the question (that is the same concept) as a different entities. This error is common when the same question appears with different identifiers in a number of branches of the questionnaire flow chart (see Fig. 1).

[1] The conceptual data model abstraction corresponds to, at a physical level, a data structure (that is a set of files/tables, records, fields) where the actual data will be collected.

Figure 1. Questionnaire flow chart: the same question appears with different codes



In a longitudinal survey, as well, the introduction of redundant entities is highly unadvisable, because it increases without need the complexity of the survey instrument and makes much more complicated re-using the collected data, in particular, to handle time dependent interviewing. The survey designer has to carefully analyze and identify the optimal data model to represent the questionnaire, in order to avoid redundancy and inconsistency. These concepts are familiar to people working with databases, but their importance has not yet been fully realized in the surveys field. A correct design ensures not only a more efficient way to store the survey data, but also better usability of collected information.

Even more than in common surveys, in longitudinal surveys the key moment is the instrument design: the conceptual modelling of the computerised questionnaire is in fact the most important moment of the entire project. The data model design is so much important in longitudinal surveys because virtually any data collected at any time could be used to customize a next interviews. Of course, only a subset of the responses will be actually used, but all of them are needed to be “imported” into current time CAI instrument. The most frequent case is the one of rules referring to responses collected at the previous (T-1) wave, but we could need to refer to the first wave (say T1) because some information are asked only the first time we interview the subject; or to a data *a priori* known about subjects (say T0); we could also want to compare the response at time T-1 with the one at time T-2; sometimes, we need to recall the last response to a question, that is the last time the respondent has been administered and has given a response to the question, time that can be variable for the different subjects. When intensive time dependent interviewing is implemented, it is very important that collected data are directly usable to set up the following interviews, without the need of any data manipulation, such as creating new tables, importing data from previous waves data files, cutting and pasting records between data files, or other manual rearranging operations. In longitudinal surveys, conceptual redundancy makes much more difficult to re-use the collected data in the next interview of the survey, and also more complicated to analyse the response data at the end of the survey.

For instance, think a frequent situation in longitudinal surveys: the interview at time T is dependent on the response to question A at time T-1. Then, we would like to be able to define a longitudinal routing rule, that will work without need of further changes at each of the following times: T1, T2, T3, etc. In order to handle this rule in a conceptually correct and straight forward way, we would like to have:

1) all the responses for question A, at any time, to be stored in the same entity A, say a data table A, along with a time identifier for each response data;

2) a grammar to define the routing logic, supporting time referencing.

Otherwise, if the CAI system we are hypothetically using does not provide such specific support, we have to arrange something sub-optimal and much more complicated, like this:

1) at each new wave, initializing a new empty response data file;

2) adding some auxiliary variables to the data structure: we need to allocate a new variable for each condition n question we want to implement;

3) importing the response data collected at time T-1 into the new auxiliary variables for time T;

4) setting up, by means of these auxiliary variables, the time dependent interviewing routing logic.

This second solution implies a huge effort in re-setting up the CAI instrument at each new wave of the survey, and requires a lot of manual work to get and re-organize the historical data involved in dependent interviewing from the previous data files. Besides more time and supervision costs, the sub-optimal solution gives higher error probability, in fact larger is the manual intervention required, more likely we can incur in manual errors.

This is only a very simple example, but it gives an idea about the importance of a correct conceptual data model design in order to set up an effective and easy to maintain implementation of computer assisted time dependent interviewing in longitudinal surveys. We outline two conditions have to be satisfied in order to develop a well designed computer assisted survey instrument.

First, the questionnaire designer has to gather expertise both from statistics and computer science, because he has to consider both the aspects of the questionnaire. Usually the researcher who makes the paper writing out of the questionnaire does not consider the conceptual modelling of the questionnaire, likely because of his lack of understanding in the computer science field, in particular the database theory. The understanding of the main principles of the information theory is a guidance for the statistician or the social researcher in the process of development of the questionnaire; we suggest they are requisite not only to computerise the questionnaire, but also to draw the paper writing up of the questionnaire. In fact, the problem cannot be ignored or bypassed by delegating a computer skilled person to computerize a paper questionnaire, because only having in mind a clear design of the questionnaire conceptual model, the researcher can make a correct paper drawing up of the questions flow, easy to computerize; on the contrary, it is unlikely that a programmer can re-design in a substantial way the researcher paper writing, even when it would be necessary.

Second, the CAI system has to give specific support for the implementation of a longitudinal questionnaire. The instrument design is highly dependent on the CAI software system available for implementing the survey project, namely the programming architecture (language-based *vs*. data-based) and the data engine architecture ("proprietary" data files *vs.* relational database). These are the two core components of a CAI system which determine how the interviewing program and the data model can be implemented. The main problem with CAI systems lacking of specific support for longitudinal surveys is that they do not allow the researcher for implementing the survey instrument according to his own conceptual design; then, he is forced to arrange some sub-optimal and possibly incorrect solution. In the following we will explain more in detail what we mean for *specific* support for longitudinal surveys and suggest some solutions for developing new CAI systems.

3. CAI support for longitudinal surveys

The most of CAI software systems currently available, including the main commercial products, have not been designed for longitudinal surveys, hence they do not provide specific support for time dependent interviewing. In fact, nothing really new has come from the latest generation of commercial CAI systems, which "ported" the previous versions of software to the modern windowed environment, introducing a lot of graphics and multimedia options, but no advanced support for longitudinal surveys. At present, a new generation of CAI system is appearing, based on web technologies, most of them as open source projects or released under some kind of public domain license. Unfortunately, these systems, while interesting for having widely introduced the support for web interviewing (Clayton & Werking, 1998; Yen, Chou, & Cao, 2004), a promise missed for a long time by the commercial products, are far from reaching the power and effectiveness of these mature systems (M Couper, 2000; MP Couper, Traugott, & Lamias, 2001; Crawford, 2002). Many commercial products (such as Blaise, CASES, Bellview, etc) are very powerful and effective systems, then carrying out a longitudinal survey by means of such systems is possible but, whether intensive time dependent interviewing is planned, it requires considerable efforts on each new wave, because you need to re set up a new survey instrument on each new wave; and a further lot of work at the end of the survey to analyse the response data, because you have to gather the data files coming from all the waves.

In our vision, the fundamental requirements of a CAI system designed for longitudinal surveys is supporting direct referencing to response data indexed by time. The CAI system has to provide effective techniques for making direct access to time indexed response data, in order to set up routing rules, consistency rules, feedbacks, etc. Such features should be based on a proper underlying data structure, designed to collect all the response data for the whole longitudinal survey, where each single response is associated with the time which refer to.

When the available CAI system does not provide such a specific technological support, carrying out a survey with intensive time dependent interviewing is quite problematic; in fact, the researcher is forced to abandon his design in order to settle something working with the available tools. The lack of appropriate longitudinal support forces the survey designer to make ready sub-optimal solutions using the available tools, wasting the best of what the computer assisted interviewing technology could theoretically allow for doing. Most researchers in the survey field seem to be resigned to the common limitations shared by the current CAI software; on the contrary, we think that better systems are possible. In the following, we present some main features that a CAI system should make available to properly support longitudinal surveys, suggesting the solutions we ideated for their implementation.

3.1Longitudinal response data base

We suggested (Capiluppi, 2002) that the core component of a CAI system for longitudinal surveys is a *longitudinal response database*, a unique data structure designed for storing the responses collected from all the waves of the survey, based on a simple idea: collecting the response data associated with their response times. A longitudinally oriented design avoids, in principle, the need of successive re-design of the data structures, and the correlated work of manipulations of data contents, in order to set up the next wave of the survey. This means that all the work of implementation for the whole survey can be done one only time, at the beginning of the survey project.

The primary requisite for a longitudinal response database is flexibility. The flexibility of the data structure is important, actually, in all kind of survey, because it is often necessary, while not advisable, due to design errors, making changes to the data model during the interviewing stage, that is after data began to be collected in the response database. Hence, it is very important to have the option of hot deck modifying the data model, without halting the running survey and without losing any already collected data. Moreover, in longitudinal surveys, this requisite is even more important, because it is likely that the researchers want to add some further questions on a following wave of the survey.

In order to comply such actual needs, the response database has to be *incremental*, that is it has to allow for straightforward inserting of new response fields/tables when new questions are added to the questionnaire, without impacting in any way on the other data collected. Then, we proposed a *dynamic longitudinal* database, based on a relational database architecture and automatically handled by the CAI system software accordingly to the researcher data model definition. To provide the highest flexibility, we suggested to have a response table for each question present in the questionnaire, collecting each response together with a time identifier (say TID) of the current wave of the survey.

This design is an effective compromise among data model optimality, flexibility, and usability of collected data. Modelling the questions as entities instead of attributes, that means, at the physical level, as tables instead of fields, solves in a natural way the matter about how to handle multiple responses as well responses from different waves, without introducing redundant entities. In this design, in fact, each (multiple) response is represented as an instance of the "question" entity, that is, at the physical level, each response is stored appending a new record to the table. This approach promises very high flexibility to the data structure and allows for incremental updates, as each question has its own response data table that is completely independent from the other questions. Then, a new response table can be safely added to the database when a new question is added to the questionnaire, even when the survey is in progress, without any impact on the response data already collected. This solution results in high computational efficiency, since data updating operations are made easy to program and fast to run: they can be implemented through basic SQL (*Structured Query Language*) statements (Bowman, Emerson, & Darnovsky, 1996; Melton & Simon, 1993)by an operational scheme:

a) delete old record(s) + append new record(s)

that is more efficient than the scheme:

b) seek record(s) + check field(s) value + update field(s)

With regard to usability of collected data, the suggested “super-normalized” design asks for doing some joins to gather the required data, a little pay out for flexibility. It seems a fairly good design for data analysis: the most common data analyses (such as descriptive statistics / frequency distributions) are in fact made easier by this design, in particular for multiple response questions[2]. Contrary to the usual way of operating in the survey field, it is advisable joining only the data tables needed, case by case, to carry out a given data analysis. The choice of having an underlying relational database as data storage engine, that we consider a key requirement for a modern CAI system, allows for querying the data collection by means of the powerful and universally known (in the computer science field) SQL language. Further, the modern relational databases make collected data easily accessible to any external program through the database connectivity technologies now available, namely ODBC (*Open Data Base Connectivity*) and OLEDB (*Object Linking and Embedding Data Base*) (Blakeley, 1997; Pizzo & Cochran, 1996; Signore, Stegman, & Creamer, 1995). This means that any statistical package, such as SPSS and SAS (Capiluppi, 2006), will be able to import the response data tables.

3.2Longitudinal routing rules

In longitudinal surveys, time dependent interviewing can require a lot of manual work on each new wave, because the researcher has to set up an essentially new CAI instrument (as discussed above). In order to handle time dependent interviewing in a more effective way, we propose to introduce a grammar to define *longitudinal routing rules*, for setting up, at design time, the complete routing logic for all the survey waves. We already introduced the use of the SQL language as standardized grammar for defining rules in a CAI system (Capiluppi, 2000b), specifying the rule condition as a WHERE statement of an SQL query, as in the following examples:

[2] Think only to want a simple one way tabulation for a question with k multiple responses: by this design, you can get the frequency distribution in a single tabulation; in contrast, if multiple responses were modelled as different fields of a single record, you should make k frequency tables and then pool it together by hand.

a) [Q12] = 1

b) [Q12] = 1 AND [Q13] = 3

where the question identifier (say QID) is enclosed in brackets.

Here we propose a time-extended SQL-like grammar, say T-SQL, that allows for defining conditions referring to historical responses in straightforward way, as easily as referring to current data. The key element of this extension is the introduction a time parameter that addressed directly each data (QID) indexed by its response time (TID), as in the following examples:

c) [Q12, -1] = 1

d) [Q12, -2] = 1

e) [Q12, T1] = 1

Both relative (c, d) and absolute (e) time referencing has to be supported: the most common rules are based on the previous time data (c) but some survey schemes did not ask each question on each wave, so the needed data to set the condition could have been asked at time T-2 (d), or could be asked only at the first contact (e).

A more difficult case happens when we need to access *the last* response of the subject, but the response time of this data is not the same for all the respondents, as consequence of highly sophisticated dependent interviewing schemes or, in a trivial way, due to missing responses. This very difficult *variable* time referencing is actually a very concrete case, thus we have to introduce a further special time identifier, specific for accessing the last response to a given question, as in the following example:

f) [Q12, #] = 1

where the sharp character as time parameter asks to the CAI system to search for the last response given to question Q12 by the current respondent.

Omitted time parameter refers to current time, then routing rules not involving time can be written as usual SQL conditions.

Obviously, the proposed T-SQL grammar has to be implemented and made available to survey designers by a compliant CAI system, that should be based on the longitudinal response database design presented above.

3.3Longitudinal consistency rules

The grammar introduced above to define longitudinal routing rules can be properly used even to settle consistency rules across times. Consistency rules can be expressed without distinction as validity conditions or as error conditions, nevertheless we prefer to specify error conditions, that fire when a data inconsistency is detected. On detected errors, the CAI program can send an alert to the user interviewing interface, and the response can be either accepted or cancelled, according the action associated to the rule. The proposed T-SQL grammar allows for defining longitudinal consistency rules, as in the following examples:

g) [Q12] < > [Q12, -1]

h) [Q12] < > [Q12, T1]

i) [Q12] < > [Q12, #]

where the current response to question Q12 is compared with the *previous* wave (g), or to the *first* wave (h), or to the *last* response (h) given by the subject. We can also compare two (or more) historical data between them, as in the following examples:

j) [Q12, -1] < > [Q12, -2]

k) [Q12, #] < > [Q12, T1]

where the response to question Q12 given at the previous wave is compared with the previous wave again (j), or the last given response is compared with the very first one.

Actually, the rules presented in this section can be used as consistency rules as well as routing rules, to configure in a straightforward way and with minimal effort an highly sophisticated dependent interviewing scheme, for instance to detect changes in any subject status and to administer them more in depth specific interviewing (Moore, Bates, Pascale, Griffiths, & Okon, 2006).

3.4Longitudinal feedback

Sometimes in longitudinal interviewing there is the need to recall and display any previously collected data, for instance to give a feedback to the interviewer or to the respondent, to remembered a response to be confirmed, or simply to reduce the “isolation effect” typical of computer-assisted interviewing. This can be made in a number of ways, for example recalling a given response in the text of a new question, or in another frame of the screen layout. For instance, we formerly suggested a feedback implementation by means of *alias questions* (Capiluppi, 2002), a special type of question that can be inserted in the page of the CAI questionnaire as a normal question, but displays, in place of the question text, a descriptive text about the recalled information, and in place of the input field, the recalled historical response(s).

The matter is how is it possible to recall historical data by means of the functions of the CAI system: common CAI system do not support referencing to historical responses by time, because their design do not represent at all the concept of time. A workaround is, again, introducing a new redundant variable, that has to be loaded with the historical data imported by tables/files of a given wave, and at last, setting up the feedback technique by means of this auxiliary variable. This work, of course, is needed for each historical data we want to use, increasing without need the complexity of the instrument and the overall redundancy internal to the survey data model.

In order to handle longitudinal feedback techniques in a easier way, without the need of introducing redundant auxiliary entities, we propose to provide direct access to historical data by means of the support of T-SQL fields in the question text, such as in the following example:

l) Hello [Name, T1], we thank you for continuing to participate to our longitudinal survey for the 6th wave…

where the Name response is recalled from the first wave, when we asked it;

m) Your last known job was [Work, #].
Can you confirm it or did you change job ?

where the Work response stores the last job change, that occurs at different times for each subject; only when the subject answers that has changed work, the Work question is administered again.

The CAI system can support this feature by implementing a pre-processor that parses the question text and substitutes the found T-SQL fields with the referenced historical data, picked up from the longitudinal database.

3.5 Answers list customization

Dependent interviewing techniques commonly require to tailor not the question itself but the answers list of the question. For instance, all the students of an university have to be administered with the question “Which courses are you attending ?” but the answer items that have to be displayed are dependent from the faculty. In common CAI systems, this customization is only possible by introducing a number of questions (one for each faculty), all with the same text but with different lists of answers, and setting up a switching logic to redirect the respondent to the question with the right list of answers, according its faculty. Such solutions has the drawback of introducing a number of redundant entities in the data model, and implies that the response data will be split into a number of different variables (data fields).

In order to handle this common need in a correct way, we introduced *dynamic answers lists* (Capiluppi, 2000b), a general solution for the customization of the answer items according the subject category, based on the SQL language. Our idea has been defining a variable list of items by means of a parameterized SQL query: the actual list of items for each subject is generated at run-time by the SQL statement. The parameters of the query represent the response data which the answers list depends from. Parameters are resolved by the CAI system, at run time, for the current respondent, retrieving the response data in the response database. For instance, the following query defines a dynamic list of courses customized by faculty:

n) SELECT * FROM Courses WHERE Faculty = [Q2]

where Courses is data table containing the list of all the courses offered by the university, classified by Faculty, and Q2 is the question about the subject faculty.

Here we suggest a further longitudinal generalization to our dynamic answers lists by introducing T-SQL fields as parameters in the query that defines the answers lists, as in the following example:

o) SELECT * FROM Courses WHERE Faculty = [Q2, T1]

where the subject faculty Q2 has been asked only at the very first interview. Obviously, T-SQL parameters have to be supported by the query engine of the CAI system.

This extension to our earlier solution allows for very flexible customization of the answers list by subject category, in a straightforward way even when the list is dependent from historical data. We outline this result is achieved in a conceptually correct way, without introducing redundant entities, hence saving a tremendous amount of work, both at design time and when data will be analyzed. The choice of using an universal standard language as the SQL avoids introducing a further new grammar, making available the whole power of the SQL engine.

4. Implementation of a CAI system for longitudinal surveys

The above suggestions are not only theory or dreams. All the solutions here presented have been implemented in a real CAI system developed by the author, demonstrating their feasibility and effectiveness. The running software, named CAPS (Computer Assisted Professional Surveys), is an innovative research system planned and designed specifically for longitudinal survey, based on web technologies and oriented to Web/CASI (Computer Assisted Self-administered Interviewing). This project of development of a new true longitudinal CAI system was born in 2006, from a former project of the same author, begun in 1996 and named CAPTOR, oriented to develop a system for standalone CASI interviewing (Capiluppi, 1996). Since 1998, the project was completely redesigned for online Web interviewing (Capiluppi, 1998), and then extended, since 2000, for CATI (Computer Assisted Telephone Interviewing) with interviewers management and telephone scheduling (Capiluppi, 2000a, 2000b).

At the state of the art, the CAPS system provides unique features in the CAI field, making available an advanced specific support for longitudinal surveys, such as: longitudinal data model, relational response database, extended T-SQL grammar, longitudinal routing and consistency rules, feedback with historical responses, dynamic answers list, and more over. It was, since the beginning, oriented to CASI, then it has a very friendly user interface, organized in pages as a traditional questionnaire, with a simple and homogeneous vertical layout. It has been designed for the Web, then it is oriented to CAWI (Computer Assisted Web Interviewing), and allows for integrating CASI and CATI methods. Further, it allows for decentralized CATI, having the option to turn into a CATI station each personal computer with a web browser and a phone line with Internet connection (such as an ADSL). It is based on a relational database engine; at present the MSJET engine proves to be perfectly adequate for surveys applications[3], and it is an open, handy format to use and transport. CAPS is not oriented to personal interviewing (CAPI), because it is an online client-server system, then the client stations need to be always connected to the CAPS server; but the increasing diffusion of wireless connection technologies will broaden its application field more and more (Schober & Conrad, 2007).

CAPS is not an open source project but an “open minded” project. The CAPS system is available to interested researchers and institutions asking the author.

CAPS and its ancestor CAPTOR were used to conduct a large number of CATI and CAWI surveys, in the social and in the market fields, most of them with longitudinal approach. Since 2000, it has been used by the survey service of the

[3] MSJET provides highly efficient data operations up to 1,000,000 records for table, and good enough to 2,000,000 records.

Department of Statistics of the University of Padova, and by other Italian research institutions such as the University of Cagliari, Nuoro, Verona, in long-lasting surveys such as the insertion of graduates in the job market, the household members trouble, the evaluation of teaching quality, etc. From all these experiences, we derive the considerations and suggestions here presented.

5. Discussion

The computer assisted interviewing technologies could open new frontiers in longitudinal surveys. In fact, the CAI method is practically the only way to carry out longitudinal surveys in practice, in particular when intensive time dependent interviewing is needed.

The most of CAI systems available have not been designed for longitudinal surveys, and do not provide specific support for time dependent interviewing. Because of these limitations, carrying out CAI longitudinal surveys requires an enormous work for setting up the survey instrument on each new wave. A new generation of survey systems for longitudinal surveys has to be developed, to exploit the potentialities of the CAI technology in this difficult field of survey.

In this paper we discuss the main features a CAI system should make available for longitudinal surveys, suggesting a number of technical solutions to manage the actual needs in a correct and effective way. All the solutions here proposed have been implemented in the CAPS research system, developed by the author, demonstrating their feasibility and effectiveness. Such a system allows for carrying out a longitudinal surveys with a fraction of the efforts and the costs otherwise required. A great advantage of the presented approach is that the complete implementation of the survey instrument can be done once, at the beginning of the survey project. The efficacy of this system in the daily survey work demonstrates the possibility and usefulness to develop new CAI systems with specific support for longitudinal surveys.

References

Atzeni, P., & De Antonellis, V. (1993). *Relational database theory*: Benjamin-Cummings Publishing Co., Inc. Redwood City, CA, USA.

Banks, R., & Laurie, H. (2000). From PAPI to CAPI: The Case of the British Household Panel Survey. *Social Science Computer Review, 18*(4), 397.

Bethlehem, J. (2000). The routing structure of questionnaires. *International Journal of Market Research, 42*(1), 95-110.

Blakeley, J. (1997). *Universal Data Access with OLE DB.*

Bowman, J., Emerson, S., & Darnovsky, M. (1996). *The Practical SQL Handbook: Using Structured Query Language*: Addison-Wesley.

Brown, A., Hale, A., & Michaud, S. (1988). Use of computer assisted interviewing in longitudinal surveys. In R. Groves, P. Biemer, L. Lyberg, J. Massey, W. I. Nicholls & J. Waksberg (Eds.), *Telephone Survey Methodology* (pp. 185-200). New York: John Wiley & Sons.

Capiluppi, C. (1996). *CAPTOR, a CASI system for evaluating academic teaching*.

Capiluppi, C. (1998). CAPTOR 2, a System for CASI Surveys on Internet. *Pre-Proceedings NTTS'98*, 103-108.

Capiluppi, C. (2000a). CAPTOR 2, un sistema di rilevazione Web-CASI. *Fabbris L.(a cura di) CAPTOR*, 163-174.

Capiluppi, C. (2000b). Il sistema CAPTOR. In L. Fabbris (Ed.), *Il questionario elettronico. Metodi e tecniche per le rilevazioni assistite da computer* (pp. 225-240). Padova: CLEUP.

Capiluppi, C. (2002). *Computer-assisted longitudinal surveys: Instrument design.* Paper presented at the *Papers of ICIS 2002 The International Conference on Improving Surveys*, Copenaghen, Denmark.

Capiluppi, C. (2006). *SAS System Handbook*. Padova: CLEUP.

Clayton, R., & Werking, G. (1998). Business surveys of the future: The World Wide Web as a data collection methodology. In R. Groves, P. Biemer, L. Lyberg, J. Massey, W. I. Nicholls & J. Waksberg (Eds.), *Computer Assisted Survey Information Collection* (pp. 543-562). New York: John Wiley & Sons.

Couper, M. (2000). Web surveys: a review of issues and approaches. *Public Opinion Quarterly, 64*(4), 464-494.

Couper, M., & Nichols, W. (1998). The history and development of computer assisted survey information collection methods. In R. Groves, P. Biemer, L. Lyberg, J. Massey, W. I. Nicholls & J. Waksberg (Eds.), *Computer*

Assisted Survey Information Collection (pp. 1-22). New York: John Wiley & Sons.

Couper, M., Traugott, M., & Lamias, M. (2001). Web survey design and administration. *Public Opinion Quarterly, 65*(2), 230-253.

Crawford, S. (2002). Evaluation of Web Survey Data Collection Systems. *Field Methods, 14*(3), 307-321.

De Leeuw, E., Hox, J., & Snijkers, G. (1995). The effect of computer-assisted interviewing on data quality. *Journal Market Research Society, 37*, 325-325.

Fuchs, M., Couper, M., & Se, H. (2000). Technology Effects: Interview Duration in CAPI and Paper and Pencil Surveys, *Developments in Survey Methodology* (pp. 149-166). *Ljubljana, Slovenia*.

Jäckle, A., Laurie, H., Uhrig, S., Essex, U. o., Social, I. f., & Research, E. (2007). *The Introduction of Dependent Interviewing on the British Household Panel Survey*: Institute for Social and Economic Research.

Kent, W. (1983). A simple guide to five normal forms in relational database theory. *Communications of the Acm, 26*(2), 120-125.

Lynn, P., Essex, U. o., Social, I. f., & Research, E. (2005). *A Review of Methodological Research Pertinent to Longitudinal Survey Design and Data Collection*: Institute for Social and Economic Research, University of Essex.

Melton, J., & Simon, A. (1993). *Understanding the New SQL: A Complete Guide*: Morgan Kaufmann.

Moore, J., Bates, N., Pascale, J., Griffiths, J., & Okon, A. (2006). Use of Dependent Interviewing Procedures to Improve Data Quality in the Measurement of Change. *Survey Methodology*, 02.

Pizzo, M., & Cochran, J. (1996). OLEDB for ODBC programmer. *Microsoft Data Access Technical Articals*.

Ramos, M., Sedivi, B., & Sweet, E. (1998). Computerized self-administered questionnaires. In R. Groves, P. Biemer, L. Lyberg, J. Massey, W. I. Nicholls & J. Waksberg (Eds.), *Computer Assisted Survey Information Collection* (pp. 389-408). New York: John Wiley & Sons.

Schober, M., & Conrad, F. (2007). Survey Interviews and New Communication Technologies. *Envisioning the survey interview of the future*, 1–30.

Signore, R., Stegman, M., & Creamer, J. (1995). *The ODBC Solution: Open Database Connectivity in Distributed Environments*: McGraw-Hill, Inc. New York, NY, USA.

Yen, D., Chou, D., & Cao, J. (2004). Innovation in information technology: integration of web and database technologies. *International Journal of Innovation and Learning, 1*(2), 143-157.