The PETRA Archive System

W. Schütte
Deutsches Elektronen-Synchrotron DESY
Notkestr. 85, D 22603 Hamburg, Germany

Abstract

Currently a new, PC-based control system is being introduced for the high energy particle accelerator PETRA [1]. All relevant data for its operation and control are sent with standardized IPX calls. Any subset of those data can be stored with an archive system at a rate of up to one Hertz.

Archivation* can be done at certain time intervals, on a specified relative/absolute value change or for values within a specific range. Archivation can be made dependent on the state of the accelerator, such as injection, and on the accelerated particle type. An example would be the storage of values of the proton injection orbit which differed from the last stored injection value by at least half a millimeter. Another example would be the storage the vacuum values that changed by at least 50%. The definition of the archivation is completely given by entries in a database. Archiving can be done in quasi-infinite tables. Tables of a defined depth are treated on a first-in-first-out basis. Both the definition of archiving and the archiving itself are done with Microsoft Access 2.0 as a database. The database is quite hard pressed, but powerful enough to accomplish such a job.

1. Introduction

PETRA is a large lepton and proton accelerator at DESY in Hamburg, Germany. It was commissioned in 1978 with a NORD-computer-based control system. These computers are no longer built. After our staff succeeded in getting the large HERA project at DESY working, a new control system for PETRA was started. This system is supposed to be a prototype for those of our other nine accelerators. Any modern control system should be based on powerful and modular industry standard products with a fast design cycle. It was decided to use Windows 3.1 PCs and a Novell 3.1x file server connected by a LAN[1]. The only communication protocol is IPX[2]. All the applications are written in Visual Basic 3.0, and even the device servers are true Windows applications. For the IPX communication a visual basic extension (VBX) was written [2]. It allows the transmission of integer and single arrays, of strings, of two special formats for hardware access and of a cycle message format. This latter allows a synopsis of the most relevant machine data. Information that is considered to be useful for many users of PETRA is sent by broadcasts. This means that the machine status is available to anyone on the network and that no special request or registration is necessary on the sending PC. The cycle message is special; it contains the most important information of PETRA, as well as the time and a unique cycle number. It is sent at a rate of one to two Hertz. Most broadcasts are sent as a direct response to the cycle message. Changing the rate of the cycle message changes the network traffic and the reaction time of the entire system.

2. Defining the Archivation Process(es)

An archiving system was built as part of this control system. Any defined subset of the broadcast data can be stored under defined conditions into the Access database by as many servers as needed. The definitions themselves are also kept in such a database. The following examples could be realized by the archivation system just by editing database tables: storage of the DC beam current every minute; storage of the last 1000 vacuum pressures that changed by at least 50%; storage of the values of the proton injection orbit which differed from the last stored corresponding value by at least half a millimeter. Each of those examples is called an **archivation**. The set of all the archivations is called the **PETRA archivation**. Each archivation is represented as a record in an archivation table that has specific relations with other tables to precisely define the details.[3] Archivations have the following information: a unique archivation name, a reference to the archivation server, the data definition, the particle type (optional), the machine status (optional), a history filter reference (optional), a data taking interval (optional), the maximum number of stored values of a particular type (optional), a reference to the data output location and auxiliary information (partly optional). We first look at how the data definition of the archivation is realized with the help of the detailing tables.

2.1 What information to store and how to get it?

Any information on the PETRA network segment is uniquely defined by an I.D., a format type and an index. The I.D. is an eight character string sent with all of our IPX messages. An example would be "PX_All" for the positions from all horizontal beam position monitors. The format would be a string of floats, one for each measured position. The index would be the i’th

---

*EDITOR'S NOTE: The author uses the word archivation throughout this paper. This word does not exist in standard English. It has been suggested that it be changed to the gerund "archiving" or the slightly pompous "data storage." Instead I have decided to keep it as used by the author and feel that the reader will understand what is meant.

[1]The other accelerators get each at least one LAN and file server of their own. Net segmentation should reflect the division into accelerators.

[2]This means that there is no IP on the Net.

[3]See figure 1 in the Appendix.

element. For example the 26th element contains the position "SL 141"[4]. The information can be most naturally managed in a master table of IPX I.D.s and a detail table with the interpretation of all indices for each format.

Additions, removal and significant change in the equipment lead to a different way of using the IPX format. These will result in different table entries. These tables only reflect the format at the time they are used and not its history making it possible for the archiving system to be able to deal with past years' data.. In other words one wants a backward compatible model for the archive data representation.

The I.D.s are not only related to data belonging together, but also to hardware implementation. For example there are 277 getter pumps in PETRA, which are treated by two different device servers, both having different I.D.s for the same types of values. One gets 93 vacuum values for the north half and 164 for the south half of the ring. The archiving system should be able to treat all those values simply as PETRA vacuum getter pump pressures. In other words one wants the ability to have a logical archive data representation.

Both arguments lead us to use a separate internal and external model for the data. All the getter pump pressures have the internal data name "GP". They are all of data type single and have the unit mBar. In a detail table there is a list of all components belonging to the data name "GP“ and an external key into the list representing the data sent by the IPX. If a pump is removed one only has to remove the foreign key entry and the corresponding record in the IPX I.D. detail table[5]. The archivation process involves selection and reordering of data.

In the internal model there are two types of data, scalar and vector. The DC beam current is scalar; at any point in time there is only a single value representing it. The 277 getter vacuum pressures are vector data. For data marked as scalar in the master table the corresponding index of the component will not be stored.

2.2 When to Store the Information?

It is not desirable to archive or even look at all the specified information each time it is sent into the network. For example, the getter pump pressures are only interesting to look at once every ten seconds. This is accomplished by an entry in the master table of IPX I.D.s. Such a strategy saves data processing power at an early stage. Some information is only interesting if there is the right kind of particle in the machine. For example, a fine grained archivation of the lifetime is necessary only when there are leptons in the ring[6]. For the particle type one has the choice of: any, protons, leptons, electrons or positrons in the machine. Some information is only interesting when the machine is in the right state. For example a proton injection orbit requires the machine to be in injection mode with current. For the machine state one has the choice between any or injection, ejection, ramp up, ramp down. Each choice can be supplemented by "with current". It is simple to implement more choices, but it requires to code changes in the archivation server program.

Assume the particle in the machine is right and the machine is in the correct state. Still one does not want to store all values at one Hertz. The simplest way to reduce the data volume is to introduce fixed intervals for storing, e.g. storage of the machine current every sixty seconds. Allowed intervals can range from one to 32,767 seconds (a little more than nine hours); "no fixed interval" can also be used. An attempt is made to read the data at the same relative point for every fixed-length interval. Data with a one minute interval is always read at the exact minute. Data with a twenty minute interval is read at the exact hour, the exact hour plus twenty minutes and plus forty minutes. Standard target times make it easy and efficient to correlate different channels, such as current, lifetime and energy, since the data taking times should be very nearly equal for equal intervals.

Fixed interval data taking has two potential problems. Firstly, no data are taken between the intervals. Secondly, some values, such as particle type, remain constant for long times. An appropriate choice of interval involves a trade-off between redundant and stale data. Therefore an alternative means of deciding when to store a value is provided. This is the so called **immediate history filter**, whereby any datum can be specified for storage when its value changes. This is ideally suited to the particle type. Numeric data can be specified for storage when the value changes by a minimum amount (good for "linear channels") or by minimum percentage (good for "logarithmic channels")[3]. A typical linear channel is a beam position; a typical logarithmic one a vacuum pressure. Also, the range of tracked values can be restricted with optional minimum and maximum values. The first time a value exceeds the limits, it is stored,[7] and no more values will then be stored again until one is read which is within its limits. **Range restriction** allows the data of a DC current archivation to be ignored, with 10% resolution, when there are no particles in the machine. The noise of the transducer is usually not interesting. Range restriction is also a low level implementation of limiting data to their physical range. The filter is implemented in such a manner that one can define a default, and then add different values

---

[4]South Left at 141m

[5] It may be the case that one is interested only in the pressures of pumps which are exposed to much synchrotron radiation. In this case one would have to invent a new data name, such as "GPhighSy" and add in the detail table the list of corresponding devices. It is possible to combine any data, as long as they are available on the same segment of the network and it have the same atomic data type.

[6]PETRA is used as a storage ring for HASYLAB synchrotron radiation experiments.

[7]Though one is by definition not interested in the value itself, it is necessary to store it (or any other non-valid value) to flag that the previously stored one is no longer valid.

for some special channels. This allows, for example, filtering noisy vacuum pumps and also being able to refer to an "UHV50%"[8] filter as a single entity in the definition of the archivation.

Another problem to solve for history filtered data is signaling that a particular archivation stopped. This might be due to the breakdown of the corresponding device server or to a pause of the archivation server itself. Here the convention is to fill a Null[9] value right after the last known valid time. This is done in the following situations: firstly, an IPX message of with a particular I.D. does not occur within a specified time-out period, and, secondly, the archivation server is shut down in an orderly manner. At start-up of the archivation server, it is checked that all entries end with a Null. If one does not the Null is added, with a timestamp, immediately following the last valid datum. Important operations, such as shut-down of an archive server, and important incidents, such as a lack of IPX messages, are logged. Normally it is sufficient to look at the data to get a valid picture of what has occurred.

2.3 Where and how to store the information?

This decision was heavily influenced by the use of Access and not using either direct files or a full featured client server database. We use a table for each archivation, the table name by definition being the archivation name, and the only choice remaining being how to divide the tables over different databases. This allows control of the size of different database files and keeps some logic in their organization. The table format consists of timestamp, data index (only for vectors) and value. The timestamp is the time at which data was taken, as given by the cycle message. It is coded in seconds starting from a fixed time. As opposed to the UNIX timestamp, it does not take into account daylight savings times, resulting in loss of one hour of archivation data each year. The one table per archivation approach is viable because all data therein are of the same atomic type. Conversely, in Access very long tables do not perform well.

Any database which is corrupted[10] is automatically repaired on restart. Any nonexistent database files or tables will automatically be created, including indices. The counting of the archivation channel entries is done with a special table at archivation start-up. This table is computed using the data themselves. Any first reference of an element will be looked up in the table and following references use counter arrays in main memory. Since all archivations use exclusive write privileges on a table, this leads to efficiency and consistency. The system dynamically adapts to an increase in the specified table depth but not to a corresponding decrease. The archivation server appends and edits records but never deletes any.

3. Performance, Tools and Outlook

During the last six months we collected roughly 75 MB of data in 35 archivations distributed over nine database files. Thus far one archivation server PC is sufficient. Ninety-two percent of all messages are handled within one second, 99% within five seconds and 99.9% within nine seconds. The archivation server generally runs reliably for weeks at a time. It is shut down for manual backup roughly once a week. As long as it is running, the database files are open and cannot be copied under Novell. In the future we will run update queries on a different server for that purpose.

A tool is also provided for viewing the data. The user chooses an archivation[11] and time scale and the tool tries to retrieve 300 corresponding data values. It is important to visualize gaps in data taking (for fixed time interval archivations) and invalid data (Null). Gaps lead to dashed lines on the display, times of no data have no line at all; the last and first valid points around a gap are emphasized by circles; values out of the plot range are treated as non-existent. Invalid values frequently appear due to the device server signaling invalid data. The archivation system does not translate private invalid values, in particular not to Nulls. Up to eight archivations can be viewed in one plot in parallel and up to ten plots are allowed. All values are correlated against time and not against each other.

Future tasks include a program interface to the archivation. It is hoped that with Visual Basic 4.0 a versatile OCX can be written. We are still looking into the possibility of using a full client server database. One advantage would be relatively easy access to the data from other computer platforms and the ability to handle large tables in a multi-user environment in a speedy way. We would consider using tables of atomic data type and adding a column with a unique archivation number.

4. References

[1] F. Peters et al. "The PETRA Control System“, to be published
[2] M. Peters, to be published
[3] Kay Rehlich, private communication.

---

[8]An Ultra High Vacuum filter that requires a relative change of values of 50% (default) and allows a reasonable pressure range (10^{-15} to 10^{-3} mBar).

[9]Special database value with a meaning of "undefined", which is different from any possible valid entry.

[10]Reasons are archive device server hard failures, such as loss of power or major Ethernet problems, in the middle of the transaction.

[11]For vector archivations one also has to specify the component (for example vacuum pump "NWL 127")

5. Appendix

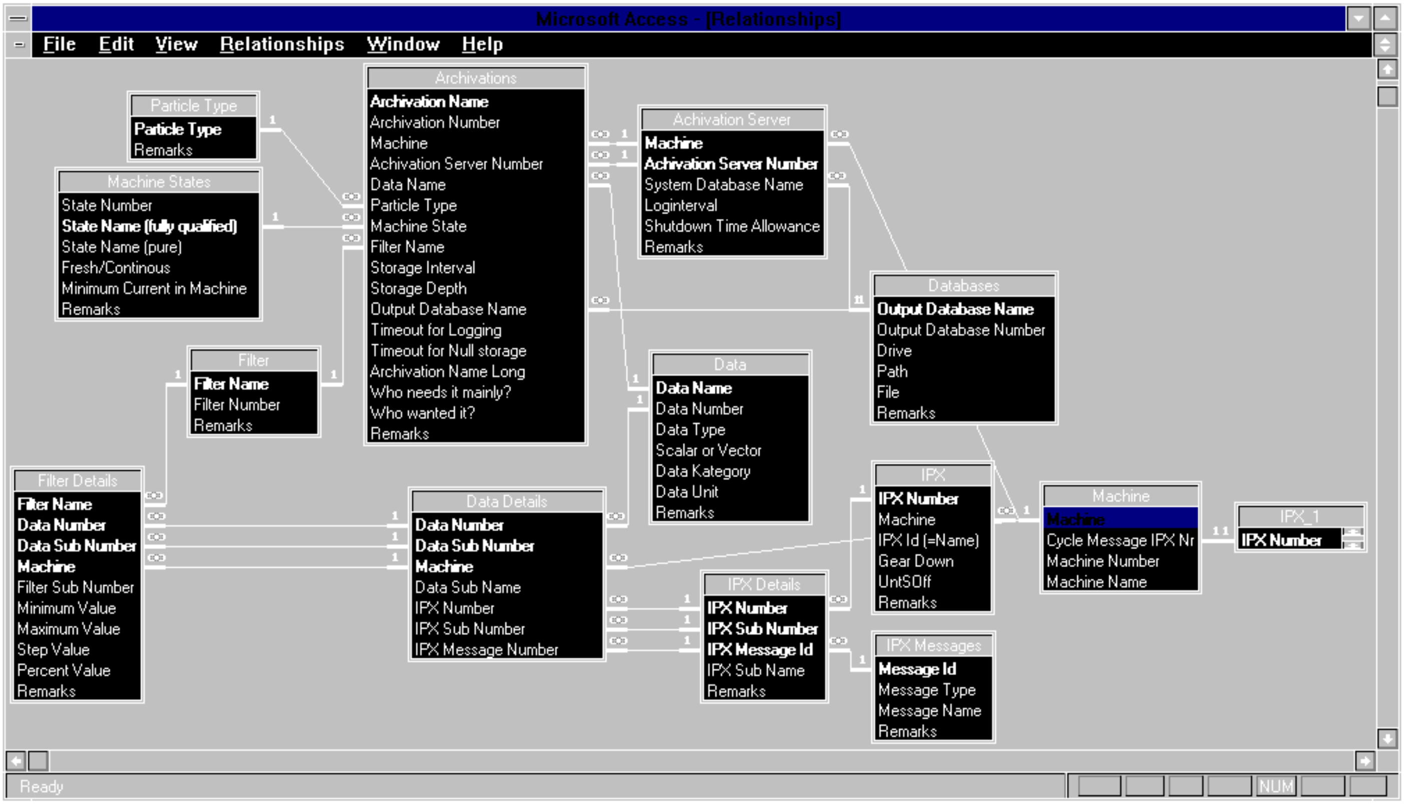

Figure 1: Definition database layout.

The central table is the archivations table. Data and data details refer to the internal data model. IPX and IPX details refer to the external data model. The filter table gives names to history filters, which are defined in the filter details table. After any change a preassembled table is created for the archivation servers (see figure 2).

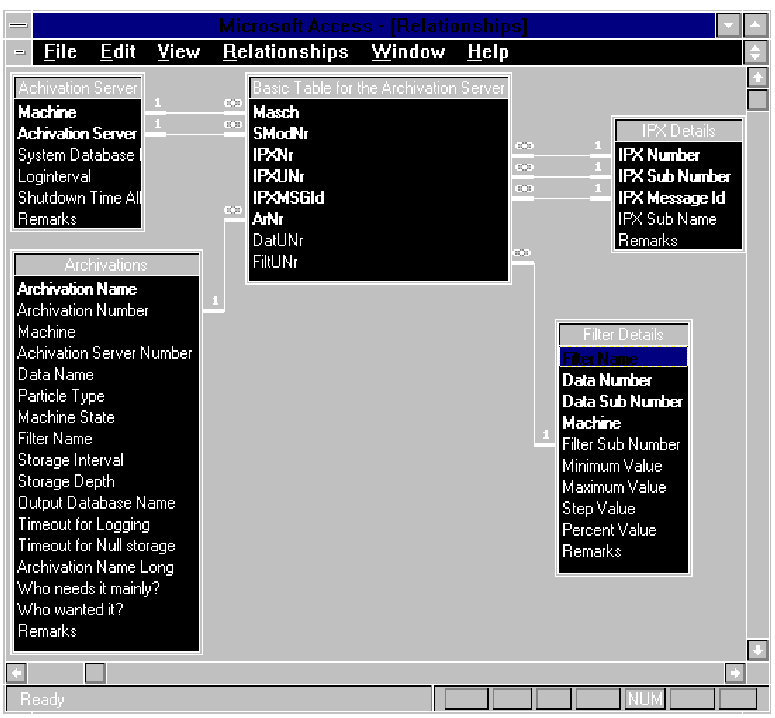

Figure 2: Definition of the preassembled table for the archivation servers.

This table is preassembled for speed of initialization. It is also easy to implement defaulting (for filters) in such a way.

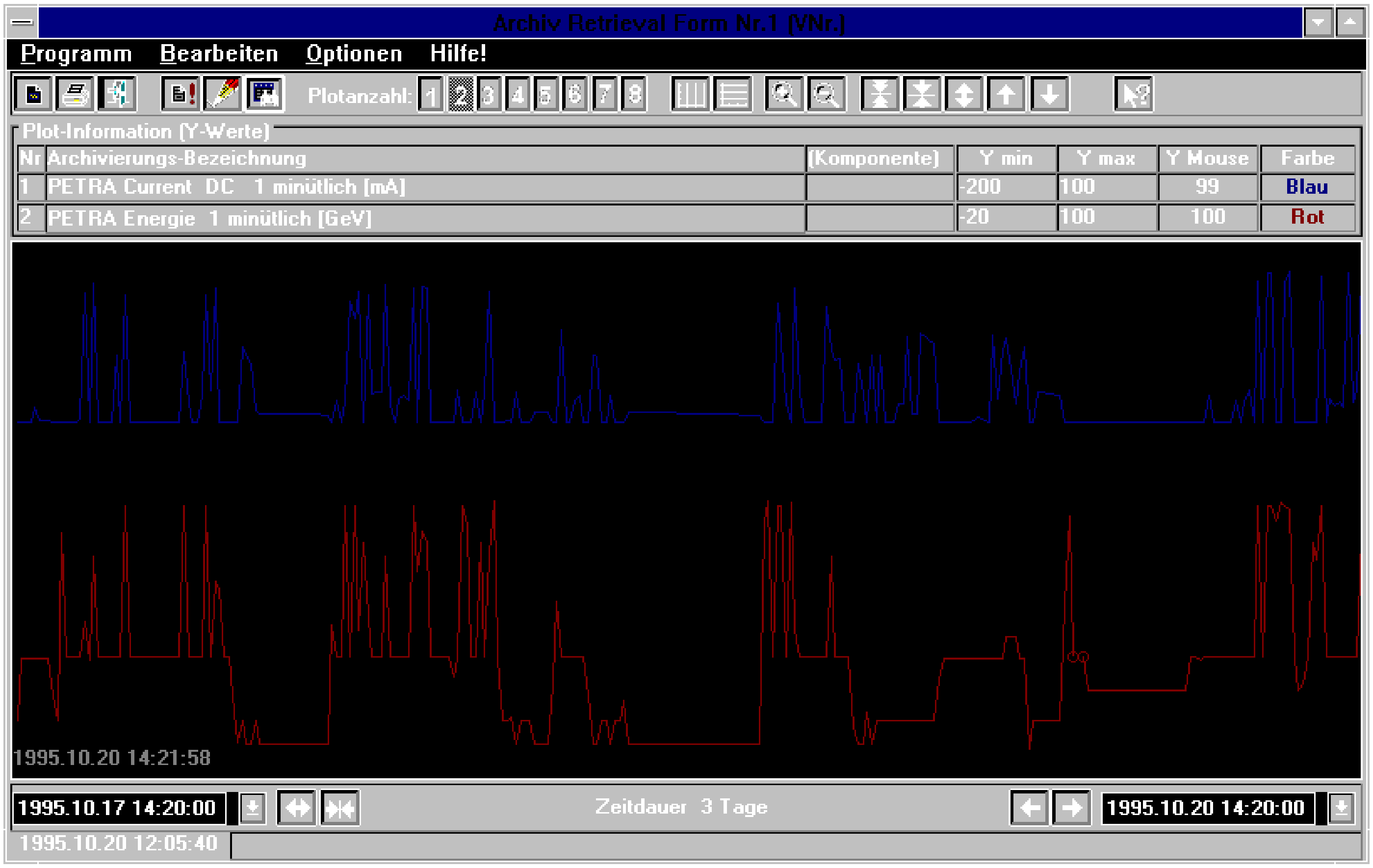

Figure 3: Viewing tool for the database.

As an example one can see the PETRA current (above) and the PETRA energy (below). The data were archived at a one minute interval and three days are shown. In the energy curve one can see a break in valid data. It is visualized by an interrupted curve surrounded with data points highlighted with circles.