Medical Informatics:
The Generic Interchange of Comprehensive Health Data

being a Thesis submitted for the Degree of PhD
in the University of Hull

by

Jeremy Charles Ellis BSc (Hons.)

October 1999

To my parents

and

Dedicated to the memory of
F.T.C. Robbins

Acknowledgements

I would like to thank all those people who have supported me throughout the duration of this work.

Thanks are due to the Medical Informatics Group, in particular Dr Penny Grubb for her committed supervision, Dr Richard Dixon for those long discussions about all aspects and Dr David Camplin for his constant support.

I would also like to thank my family and Claire for their encouragement throughout.

Publication List

D. Ingram, D. Lloyd, D. Kalra, T. Beale, S. Heard, P. A. Grubb, R. M. Dixon, D. A. Camplin, J. C. Ellis, A. M. Maskens. GEHR Deliverables 19, 20, 24 - 'GEHR Architecture', Version 1.0, 30/6/95.

J.C.Ellis, R.M.Dixon, P.A.Grubb 'The EDIFACT standard NHS Pathology Message is not Standard EDIFACT' Medical Informatics Group Report series - 1996 ISBN: 0 85958 911 0

J.C.Ellis, R.M.Dixon, P.A.Grubb 'EDIFACT Right Track, Wrong Vehicle' - EMEDI Newsletter Issue No. 11, August 1996.

P.A. Grubb, D.A. Camplin, J.C. Ellis and A.A. Takang "Medical Audit and Quality Assessment Require Technological Support." In Conference Proceedings 1996 IEEE Engineering in Medicine and Biology 18th Annual International Conference, 31 Oct. - 3 Nov. 1996 (Amsterdam) (ISBN 90-9010005-9[CD-ROM]).

J.C.Ellis, R.M.Dixon, P.A.Grubb 'Towards 21st Century Data Transfer' Proceedings from Towards an Electronic Patient Record Europe 1996. November 1996 ISBN 0 9640667 8 5

J.C.Ellis, R.M.Dixon, P.A.Grubb, D.A.Camplin, 'The Integration of data in existing systems with systems of the future', Proceedings from Toward An Electronic Patient Record 1997, Nashville, Tennessee, USA. ISBN 0-9640667-9-3

"Proceedings 2nd EU-CEN Workshop Electronic Healthcare Record" Porto Carras, Greece 25-27 May 1997.

R.M.Dixon, J.C.Ellis et al, 'The Good European Health Record Object Model' Proceedings from Toward An Electronic Patient Record 1997, Nashville, Tennessee, USA. ISBN 0-9640667-9-3

J.C.Ellis, R.M.Dixon, D.Kalra, D.Lloyd 'The Next CEN Standard for Electronic Patient Architecture' Presented at 'Towards an Electronic Patient Record 97' 19th October 1997

Chapter 1

1 Introduction and Background

1.1 The Objective

The objective of this project was to study the area of generic transfer of comprehensive medical data.

The work presented in this thesis had as its main premise the belief that generic transfer of comprehensive medical data will help towards the goal of better healthcare particularly in an environment of shared care. It studied the main methods of data transfer available at present, and as a result carried out an in depth review of one such method adopted by the *National Health Service* (NHS). Criticism of this method was made. These criticisms lead on to the development of an alternative method of generic data transfer based on an emerging European standard for the storage of medical data. This in turn led on to the consideration of data in legacy systems. Finally, an evaluation of the developed method was undertaken.

1.2 Setting the Scene

The use of computers in everyday life has become ever more prevalent over the last ten years. The average computer on the office desk has thousands of times more processing power than was available to the scientists who put man on the moon. However man is still struggling to harness this power to store and manipulate healthcare information effectively.

Despite the huge expenditure on information systems in the NHS the information available remains poor [Holl94]. It has become increasingly obvious over the last few years that standards need to be introduced into healthcare computing to exploit the large amount of information that healthcare related systems hold. A key to the exploitation of such information is the communication of the underlying data in a generic format that all electronic information systems can understand. At present there are many independent methods of passing data. These are generally point-to-point transfers and are specific to the particular medical information systems on which the data is recorded.

The interchange of electronic data between different sites was at first seen, by institutions, as a means to gain advantage over competitors [Grah94]. There may have been some short term gain from this but in reality the exchange of information electronically has to be viewed as collaborating with partners to capitalise on the

reduction of areas such as operational costs. In order to exploit this kind of data transfer fully it needs the co-operation of all parties that are involved in it. This has become even more apparent as non-profit making organisations, such as the NHS, have started to utilise electronic data transfer.

Although no study has been carried out in Europe to evaluate the amount of money saved by communicating health data electronically one has been undertaken in America. The results show that an estimated saving of 30 million dollars is accrued every year [Hosp95]. This highlights just one of the areas of savings that can be made. It is envisaged that the use of electronic messaging techniques for the transfer of data will improve patient care by providing the means whereby information about patients is available when and where it is needed [Hosp95].

The NHS has started to adopt structured messaging techniques as part of a global strategy in the form of the NHS-wide network [Im&t94]. The overriding objective of the NHS-wide data networking strategy is to ensure that 90% of NHS organisations are able to exchange data electronically if required. Although there is defined an overall strategy, the tactics used to achieve this goal appear to be fragmented, with several localised projects being carried out in each region [Im&t94]. In many cases these projects are re-inventing the wheel, consequently wasting money, resulting in the same work being carried out which cannot be integrated into a single common architecture without a great deal more work.

1.2.1 Why EDI is Needed

To gain the maximum benefit from existing Healthcare data it needs to be available to the practitioners, researches and managers that can use it most effectively. Due to the nature of primary and secondary care within this country these practitioners are likely to be in separate institutions, which means that the medical information needs to be transported. The quickest way to facilitate this is to store this information in an electronic format and transfer the data using a communication infrastructure, irrespective of whether this is by landlines or via radio signals.

1.2.1.1 The Human Factor

It is a fact of life that the world in which we live today people are becoming more mobile, travelling further and expecting faster service in every aspect of living. As a

result of this the systems that are in place for the provision of medical information have become outdated and inadequate. The days of storing patient information in Lloyd George envelopes and sending medical information by post will shortly be gone. Information is needed in the main at the point of care of a patient. Whether this is for a patient that has become ill while on a business trip to the opposite side of the world, away from their medical notes, or for a patient who is undergoing treatment at different institutions of care. This means that methods of storing and communicating electronic data have to be in place as the world becomes progressively smaller.

Many of the early systems that provided transfer of data required human intervention, which has been viewed as a disadvantage [Kay93]. To encourage the use of data exchange systems that can be of use to medical staff, systems will be required to have little or no manual intervention. Systems must fit neatly into the workflow and not discourage use by being overly labour intensive. If this is not adhered to then there will be a resistance to change that could result in systems not being used. The quality of the data that is being received also has to be reliable before staff will be confident in using such systems [Dixo98]. These are very real human issues that have to be addressed.

1.2.1.2 Epidemiological Issues

The aggregation of patient healthcare information for epidemiological studies to predict trends in illnesses and research in to the causes of diseases is another area that will benefit from the exploitation of data transfer in a generic way.

1.2.2 Potential Problems

Computerisation of different areas in the past has shown that commercial interests have been served by keeping systems developed in isolation.

> *"There is little or no standardisation of the record structure between systems; indeed incompatibility has in some instances been deliberately sought to protect a share of the market"* [GEHR95, p26]

The most popular system (often provided by the company with the best advertising department) then becomes the leader, forcing *de facto* standardisation through marketing e.g. IBM with respect to hardware standardisation and Microsoft with respect to software. The commercial interests of other solution providers then become

best served by conformance with this *de facto* standard. However, in the interests of better healthcare, standards should be arrived at by consensus rather than led by market forces. This may be forced by legislation or encouraged by national bodies such as the NHS. With guidelines like the *Requirements for Accreditation* [RFA95], produced by the NHS in the UK, that defines requirements to which electronic healthcare systems have to conform. Another scheme introduced by the NHS is *Information for Health* [Lang98], whose aims include delivering:

- "lifelong Electronic Health Records for every person in the country",
- "integrated care for patients through GPs, hospitals and community services sharing information across the NHS information highway".

1.2.2.1 Capturing Patient Data

Data is being transferred between systems for a variety of reasons and in a variety of ways. Without an underlying information model, there is a danger of compromised integrity. The main problem areas are outlined below.

Take the example where a healthcare information system has recorded the following data during an encounter with the patient:

Weight : 76 kg
Blood Pressure : 120/80
Tumour
Size : 3 cm
Location: Lower Abdomen

Table 1

It may be thought that *size* or *location of tumour* could be transferred to another site as part of an agreed data set. A number of systems and projects have attempted to achieve this by such means as:

- *writing a single piece of text to a file*

  When the sender and recipient have agreed the item of data to be transferred and its position in the file, the name of the item (*size of tumour*) may or may not be sent with its value. Often the units (*cm*) will not be specifically sent but be assumed by default.

If this is done on a regular basis and the sender/recipient changes any of these data (such as units) without prior modification of the process at both ends the data can be open to potentially dangerous misinterpretation. For instance, a tumour of size 3 cm is very different from a tumour of size 3 inches.

A further problem may arise as a result of this item of data having been taken out of its immediate context. Other information such as *weight* and *blood pressure* that may not be relevant to the sender or recipient at the time may become of vital importance in relation to other observations.

Additional facts such as when the data were observed, who was responsible for the data and where they originated may later prove to be important for example in a case of litigation.

- *capturing the value on screen*

  In capturing data via screen dumps, there is a possibility, particularly with non-GUI based systems, of locating the required items name on the screen display generated by the system. This may well rely on the data to be captured remaining in precisely the same place relevant to some other data on the screen or on interpreting the screen representation of the name. As in the case of writing data straight to a file, the context will almost certainly be lost, indeed it may be difficult to determine in this way what that context was and thus there is further opportunity to cause loss of integrity. Capturing the data automatically as it is sent to a hard-copy device also exhibits all these problems.

- *e-mailing the value*

  This is slightly better but by no means complete. For instance, it is possible to record automatically the date and time it was sent as well as who sent it. The data extracted for use in the e-mail is liable to exhibit the faults already explained. How is the data extracted for sending via e-mail and how is it integrated into the receiving system? If this is performed automatically, where does the responsibility lie? [Dixo98] Manual re-keying is open to the possible introduction of errors.

- *use of structured but inflexible transfer protocol e.g. EDIFACT, HL7, ASTM1238*

  Obviously this is an improvement as it includes the attributes needed at any particular time and encourages the recording of contextual information where it

conforms to the prescribed protocol. The use of recognised codes reduces the chance of misinterpretation of data. The transfer protocol may also include some mechanisms for security however there are several drawbacks, such as:

- the prescriptiveness of the messages [Elli96a] where only the items that have been determined in advance can be transferred between cooperating sites, this is discussed in more detail in Chapter 2. This leaves open the possibility of vital information and/or vital context being lost
- message types can only be adapted or extended after a lengthy process of consultation, in an environment of constant change [Elli96a]
- as with e-mail, how is the data captured and retrieved?

Another recording system to consider is the paper based one. Paper notes contain a wealth of information, but in order to use the information to its potential, the full richness of the data needs to be recorded electronically. Any system that attempts to do this via inadequate methods will again be subject to problems. For example, the paper information could be kept as scanned images but this does not allow the data to be used or processed. It is no more accessible than when on the printed page. Optical Character Recognition (OCR) software does allow the conversion of a scanned page to text. However, unless 100% reliability in conversion can be guaranteed, the quality of the captured data must be questionable. Even if capture can be guaranteed to be 100%, the relationships of the text (i.e. its semantics) will still be absent.

Natural Language Processing (NLP) holds out the prospect of being able to convert free text and even speech to a structured form in real time - combining freedom of expression for users with structured database storage. Pre-defined dictionaries are used to allow automatic indexing of tracts of free text. NLP is likely to be useful first in information retrieval. The arguments in favour of NLP are strong but at present NLP requires much processing and at this stage of development is probably not really an alternative to structured storage.

Interlingua - an artificial language between natural language and coding systems to manage translation between pairs of languages - has been suggested by some authors as a possible way forward [Gang92]. This concept is not yet fully developed - although should be kept under review. For example, groups such as Galen [Rect95] are working towards the possibility of exchanging data while retaining maximal

expressive power and correct reflection of meaning - via a formal representation of medical data and knowledge to serve as an interlingua.

The problems listed in these examples are not minor nuisances that can be corrected easily, but fundamental obstacles to the integrity and reliability of the data being turned into useful information and knowledge about the patient and subsequently transferred between different parties. Since data needs to be maintained for a considerable period of time (up to periods in excess of 100 years), it is essential that integrity be maintained in order that the data remain of use. This puts the ad hoc methods into perspective.

It has been shown that none of these methods are adequate for the transfer of existing data [Elli96a]. Consequently, they will be entirely unsuitable for integration of data into the medical records of the future [Elli97].

1.2.2.2 Communicating Incorrectly

It could be argued that communicating data incorrectly is worse than not being able to exchange information in the first place, as this could lead to an incorrect diagnosis being made or artificially skewing a trend. Methods have to be set up and maintained that ensure the integrity of the data that is to be transferred at all times.

1.2.2.3 Additional Information

In addition to the importance of healthcare data being stored correctly on an information system it is vital that the information held can be traced to a clinician taking responsibility for that particular entry. As well as the data being attributed to the author extra information should be recorded with the entry, such as the date and time it was recorded. This contextual information will also have to be transferred whenever the associated data entry is communicated.

It is important also that any amendments to entries in the patient record be recorded. For example when somebody corrects a mistake that they have made whilst entering the data the original entry must always be available to the clinicians who have access to that data. The contextual information and related information is also important from a litigation point of view. Again this information should be communicated at all times [Dixo98].

1.3 Messaging Formats

There exist many different messaging formats for the exchange of data. These are explored more thoroughly in later chapters, however a brief overview is given in this section serving as an introduction.

1.3.1 Exchange Formats and Interoperability

Two basic methods for the facilitation of data transfer are Exchange Formats and Interoperability. These methods of data transfer can be compared to asynchronous and synchronous exchange. Exchange formats are used where there is normally some intervention in the actual transfer of data such as triggering off the modules for exchange i.e. when an asynchronous connection has been made. Interoperability can describe the communication of data when no intervention is needed. When a synchronous connection has been made. However, interoperability may also be asynchronous.

1.4 Legacy Upgrades

With the introduction of standards both for storage and transfer of data consideration will have to be given to data that is held in existing systems, or legacy systems, so that the data will be conformant to the standards that are being introduced. Issues that have to be faced include: How to transfer data between legacy systems and standards based systems, how to add contextual information to the already stored data, how to make sure that the data will not be retrospectively changed at any point in time.

1.5 The Way Ahead

At the time research on this thesis was undertaken the *Good European Health Record* (GEHR) project had produced a model for the storage of healthcare data. The *Comité Européen de Normalisation* (CEN) were just starting work on the definition of a standard healthcare architecture and work was in an embryonic stage. CEN were taking on board the ideas of GEHR. The *International Standards Organisation* (ISO) was also just starting to take an interest in the area. At this stage work on the actual transfer of the held data was yet to be embarked upon at an International level.

Health information systems will need to be based on standards for medical record architectures, such as those produced by organisations like CEN and ISO, and it will be vital to have an adequate underlying information model. In the wider healthcare context there is a multitude of users using different applications, storage modalities

and computer platforms (hardware and operating systems). The applications reflect many views and uses of information. It is important that the *structure* of the underlying information is modelled rather than any particular *view* of it. The data may be viewed in an infinity of ways but the underlying information structure will remain the same. By using a standard information model for healthcare, the data will be distinct and separable from the applications that use it. Any attempt to define a standard medical record architecture must be able to accommodate the current growth towards systematisation of medical knowledge. It must support all the processes of clinical care and requirements for access to information, taking into account the wider needs for communication.

Clinical data contains a wide diversity of data types and apparently simple elements of healthcare information can at times require quite complex recording structures [GEHR92 ch. 5, GEHR95 ch. 3]. The range of methods for conveying information is not static and indeed will evolve as medicine itself progresses. The GEHR architecture provides for the recording of data of any type (from coded text to multimedia) for any observation as required by the clinician at the time of recording. Many classification systems are used in healthcare and a shared healthcare record must allow the use of any or all of these. This includes the case of integrating existing data on less flexible systems into those of the future.

Systems of the future will have to be comprehensive, portable and communicable as we go towards the 21st century.

Chapter 2

2 EDIFACT

2.1 Introduction

This chapter takes an in depth view of an exchange format for the transfer of data. This exchange format is the *Electronic Data Interchange for Administration, Commerce and Transport* (EDIFACT) standard, which was adopted by the NHS for the transfer of medically related data. One particular NHS EDIFACT message is studied and conclusions drawn that relate to all NHS messages in general.

The chapter then moves on to looking at using EDIFACT as the transfer syntax for transmitting data based on the GEHR Object Model (see chapter 4.4). Conclusions are drawn and finally a header message containing contextual information about data being transferred is defined using the EDIFACT syntax.

2.2 EDI Overview

"EDI, Electronic Data Interchange, is the interchange of standard formatted data between the computer application systems of trading partners, with minimal manual intervention." [Ecde91]

EDI aims to dislodge the paper trading cycle between business partners and instead incorporate transactions electronically. The benefits of this are the reduction of high operating costs, the saving on time and a much reduced error rate in transferred data with comparison to the paper trading cycle [Eced91].

Several EDI standards have been developed over the last few years. However there is now a move towards the single standard, *EDIFACT* (ISO 9735). EDIFACT, as has already been explained, is an acronym for *Electronic Data Interchange For Administration, Commerce and Transport*. This is an international standard format for the interchange of data; it helps to overcome the complications that can easily arise when a non-standard message passing approach is used.

Standards are necessary within EDI to provide a suitable means of communication that every system can understand. Without a common language there is chaos. As there are so many ways of transferring data it may be costly and time consuming to interpret those messages in different formats. A different interpreter would be needed for each different form of data transfer. Between small numbers of trading partners this may be an acceptable way in which to work. However, with progressively more partners, the conversion process becomes unmanageable.

Figure 1, below, shows the number of transfers of different data translations between five partners each using a different form of data transmission.

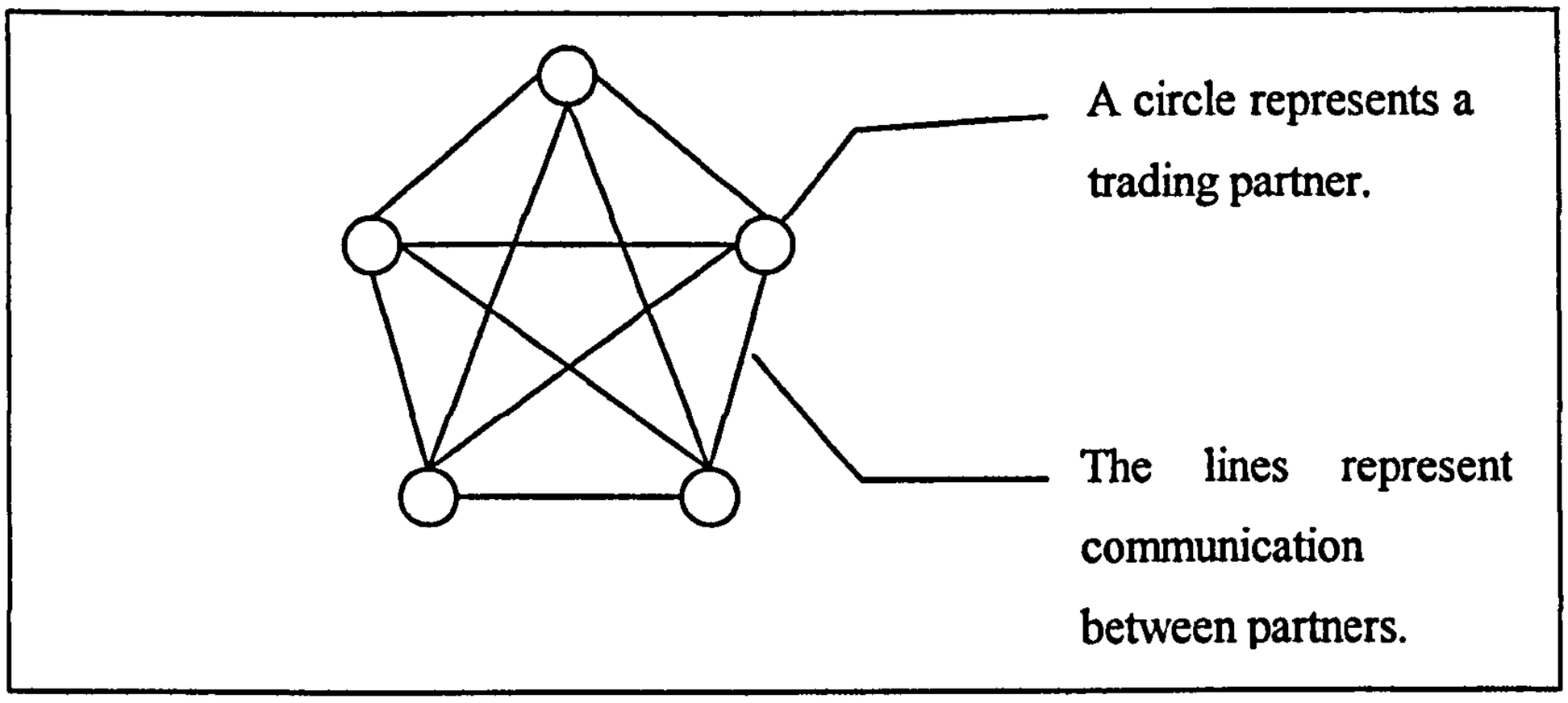


Figure 1

Assuming that the format for sending information and receiving it is different. There could be as much as twenty different conversions taking place between five partners. The general equation for calculating the number of conversions taking place between *n* partners is shown below

$$2\left(\sum_{x=1}^{x=n-1} x\right)$$ Where n is the number of partners

It can be seen from this that with one hundred partners the number of conversions becomes very large. Bearing in mind the thousands of partners involved in modern commerce, the scale of the problem becomes apparent.

However with one common interpreter it would become less time consuming, less expensive and less confusing to transfer data. This is shown in Figure 2.

As can be seen from the diagram, each partner only has to interpret data in one format.

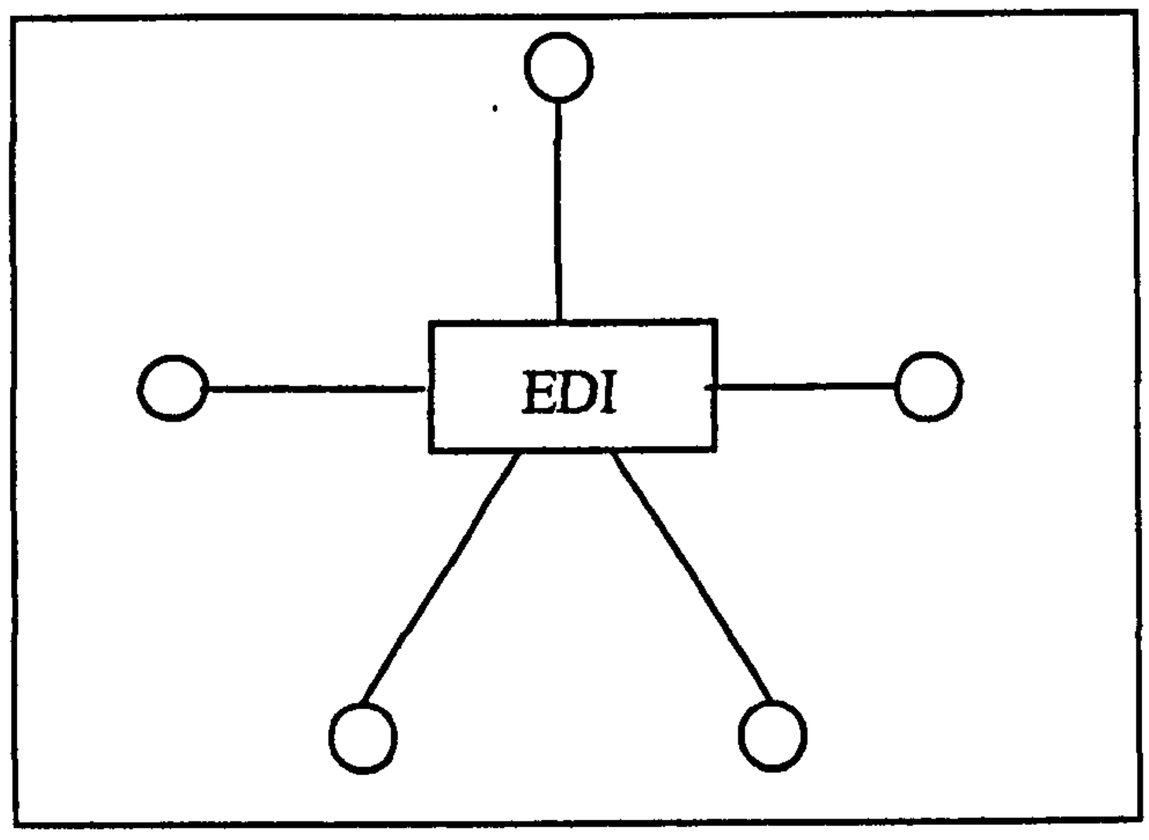


Figure 2

The EDIFACT standard consists of a grammar (syntax and rules for structuring the data) and a vocabulary. The vocabulary is contained in directories that take the following format:-

Data Elements

Segments

Messages

After much consultation with the NHS, professional and commercial organisations, the NHS Management Executive decided to adopt EDIFACT as the NHS standard for the electronic format for the exchange of structured messages [Dohl92]. This standard has been adopted for information exchange between the NHS and external organisations as well as internally.

2.3 EDIFACT Structure

2.3.1 EDIFACT in Detail

This section introduces the construction, terminology and definitions associated with the EDIFACT format for data transfer. It is essential to understand the format in order to design and implement EDIFACT messages. The EDIFACT interchange can be represented as a hierarchical structure. Figure 3 shows its components and their relationship with each other.

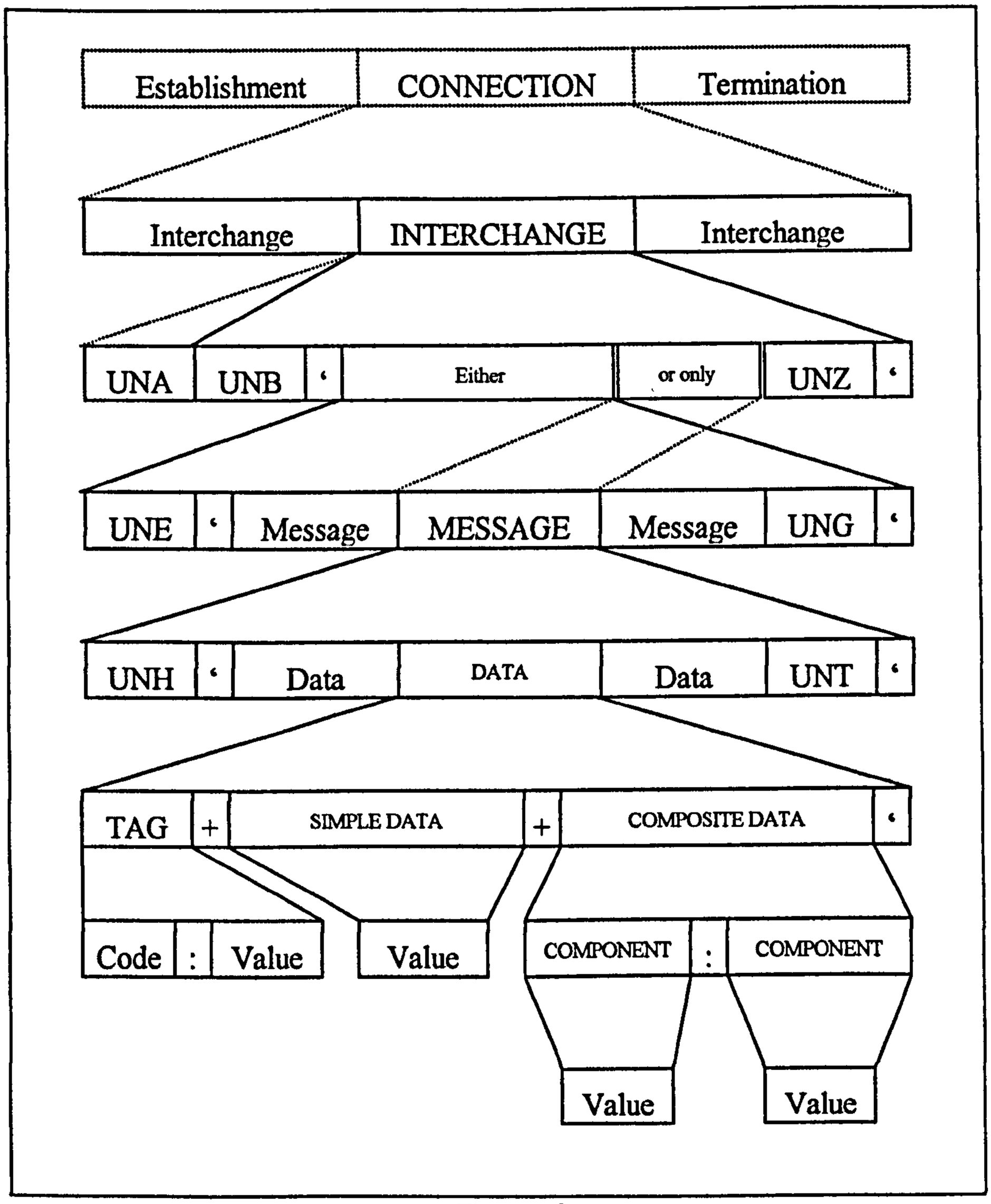


Figure 3

Each part of the interchange is considered separately within this section.

2.3.2 Data Elements

At the very lowest level of a message are the data elements. They are the smallest part of an EDIFACT interchange. Data elements identify an individual field or item of data designed for a specific purpose, such as a unit price or measurement.

There are two types of data element that can be described within a message. These are: -

- *Simple Data Elements*

These identify single items, such as postal code, age or patient forename. A simple data element can look like this: -

Tag No.	Description	Status	Rep.
3818	Patient Forename	C	an..17

Table 2

- *Composite Data Elements*

These are formed by a combination of two or more component data elements. For example a composite of Patient Forenames can be made up of several repeats of the simple data element, Patient Forename.

A composite data element looks like this: -

Tag No.	Description	Status	Rep.
C946	PATIENT FORENAMES	C	
3818	patient forename	C	an..35
3818	patient forename	C	an..35
3818	patient forename	C	an..35

Table 3

This composite is made up of three occurrences of the simple data element 3818. The format and contents of both data elements are explained in the next section.

2.3.3 Data Segments

A segment contains the transaction information held within individual data elements. Directories of pre-defined data segments exist for use in the health arena. The data segments consist of logically related composite data elements and/or simple data elements fulfilling specific functional requirements, such as name and address. The structure of segments within EDIFACT messages is designed to be flexible. Some segments are mandatory (i.e. they have to occur within a message) and some of them are conditional (i.e. they do not have to occur in a message). The same segment may occur several times within a message.

A typical segment that occurs several times in a message is the Date/Time/Period segment, which is used with different values within the message.

An example of a segment is the Patient Personal Identification, which is a segment designed for use in the health arena :-

PPI - PATIENT PERSONAL IDENTIFICATION Function: To provide structured personal identification information for a patient.			
9809	PATIENT CATEGORY, CODED	M	an1
9801	SEX, CODED	M	an1
3802	PATIENT FAMILY NAME	M	an..35
C946 3818 3818 3818	PATIENT FORENAMES patient forename patient forename patient forename	C C C C	 an..35 an..35 an..35
3824	PATIENT NAME TITLE	C	an..17
3804	PATIENT PREVIOUS FAMILY NAME	C	an..35
3804	PATIENT PREVIOUS FAMILY NAME	C	an..35
3804	PATIENT PREVIOUS FAMILY NAME	C	an..35
3822	PATIENT NAME SUFFIX	C	an..17
C970 9811 1131 3055 9810	MARITAL STATUS marital status, coded code list qualifier code list responsible agency, coded marital status	C C C C C	 an1 an..3 an..3 an..35

Table 4

The PPI segment comprises both kinds of data element. The individual data elements within the PPI segment have numbers preceding them such as (3818) patient forename. These are known as tags, and are a unique description assigned to that data element. The tags starting with C, such as (C946) Patient Forenames, denote that the data element is a composite. The individual elements that make up a composite are known as component data elements (See Figure 3).

Each data element is shown to be mandatory or conditional by the M or C that follows the element name. If the data is mandatory then data must appear in the element. If the element is conditional then the inclusion of data during usage of the message is

optional. Each segment also has a mandatory or conditional status within a message, with the same rule applying.

Each data element has a data value representation shown on the far right hand side. The representation of data may be alphabetic, numeric or alphanumeric, as follows: -

Representation:

a	Alphabetical characters
n	Numerical characters
an	Alpha-numeric
a1	Alphabetic fixed length 1
n3	Numeric fixed length 3
a..3	Up to 3 alphabetic characters
n..3	Up to 3 numeric characters
an..3	Up to 3 alpha-numeric characters

The data segments are of no fixed size but each new segment designed to contain patient information has to be ratified by the *UN/EDIFACT* ratification board.

2.3.4 Messages

A message incorporates a selection of segments to make up a specific business transaction. These messages correspond directly to a function, such as invoices or purchase orders, and contain information relevant to that function. In order for the message to be understood without ambiguity the interchange requires the implementation of rules and syntax.

Messages therefore have to be structured so that the contents of each message make sense. A message, as can be seen in the hierarchical structure (Figure 3), is made up of data segments that are in turn made up of data elements.

Many messages of the same type make up what is known as a functional group, where all messages transferred are of a similar subject. A combination of these functional groups and messages make up the final interchange.

One of the advantages of transferring data in the EDIFACT format is that it checks the integrity of the data that is being transferred. The service segments, shown in the hierarchical diagram as UNA, UNB, UNG, UNH, UNT, UNE and UNZ do this checking. Each of these segments forms the header and trailer of a message. The header contains reference information and the trailer contains terminating and error checking details.

2.3.5 Branching Diagrams

A branching diagram is the graphical hierarchical chart that shows the structure of a message. It shows the segments that are used, whether they are mandatory or conditional and the number of times that they may be repeated within a message.

The highest segments in the chart are service segments or non-repeating data segments. They are located at level 0.

Level 1 and higher numbered segments are either repeating data segments, or segments that have beneath them hierarchically related segments, these segments are often grouped.

The structure of a branching diagram can be seen in Figure 4.

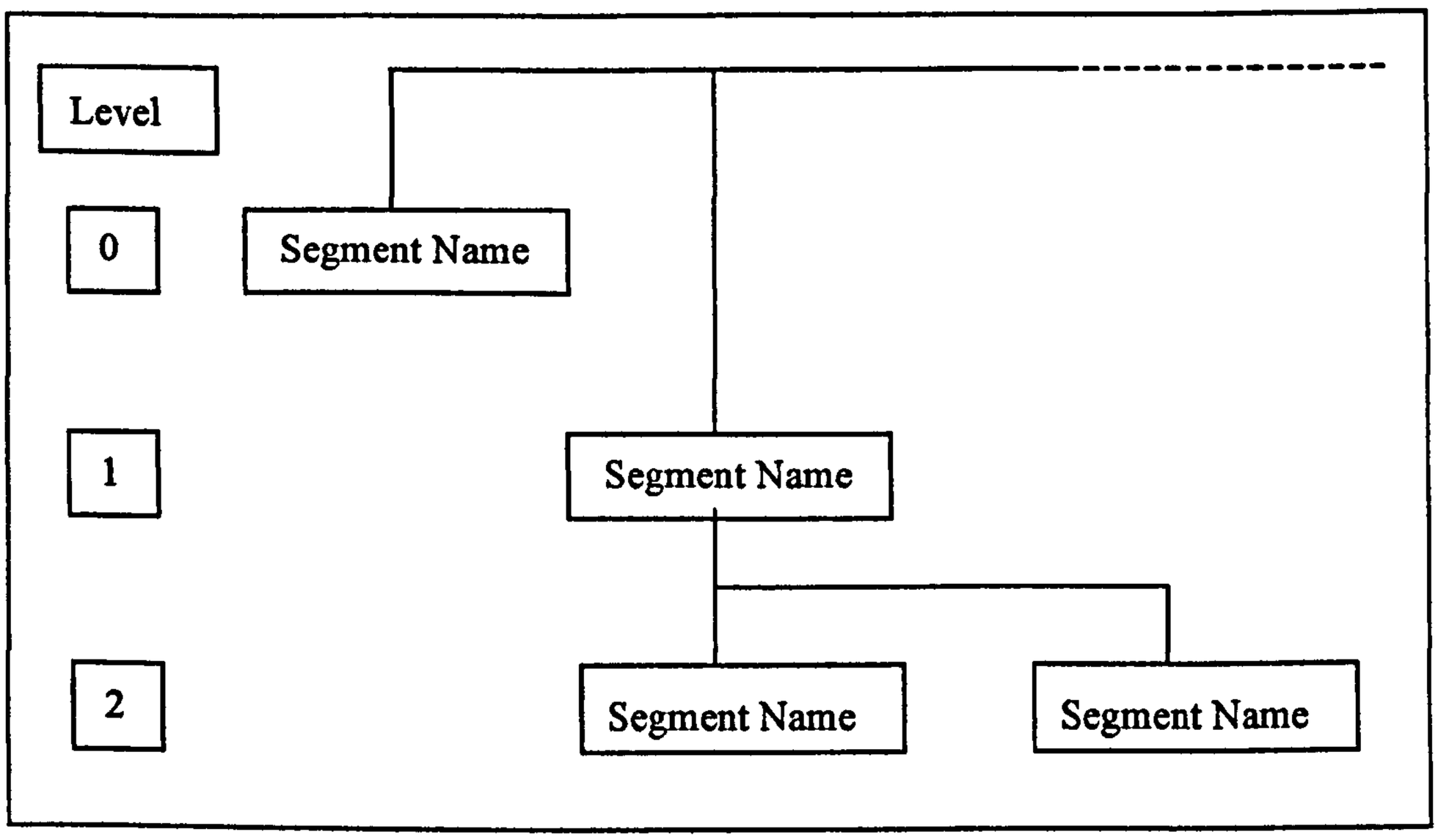


Figure 4

2.4 European EDIFACT Healthcare Messages

It is generally accepted that EDI is desirable in the healthcare sector as it enables patient centred information to be available at the point it is needed. To this end, many attempts have been made to transfer information in different formats. To bring some continuity to this area a report was produced by the *Comité Européen de Normalisation* (CEN) *Technical Committee* (TC) 251, CR1300, in 1992 that stated:

"A design method for healthcare messages should be developed which is independent of the target syntax"

Another report produced by CEN TC251 [Inve93] specified such a message development methodology that was independent of a particular syntax. It provided for this by designing a *Domain Information Model* (DIM) which is a conceptual model encapsulating the problem domain of the area being represented. An intermediate step is then taken between the DIM and the message syntax, which is known as the *General Message Description* (GMD). One GMD may be seen as a special view of the overall DIM reflecting one message type. This GMD can then map onto any exchange syntax to facilitate the exchange of messages. This is a good idea as it stops the end message that is developed being restricted by the syntax that is to be used.

It is suggested, by the author, that in addition to the findings of the CR1300 report, the structure of the messages needs also to be independent of the precise contents, i.e. the data fields that are to be transferred are not prescriptive. This is not catered for by the method adopted by CEN. As well as a new GMD being needed for each specific message in a group, a new DIM is required for each group of messages. That it would be far simpler if there were one model encompassing all data fields that are to be passed is self evident, but has not in the past been considered possible.

Although the design model for the transfer of messages should be independent of a syntax, a syntax for the actual transfer of data is needed onto which the messages, when designed, can map. The exchange syntax that has been used by CEN and adopted in the UK [Dohl92] is EDIFACT, which has been successful in areas of trade since 1987.

2.5 National Health Service EDIFACT

The National Health Service (NHS) adopted EDIFACT in 1992 [Dohl92] for the transfer of data pertaining to healthcare.

Early in 1995 the *Information Management Group* (IMG) of the NHS Executive released a number of messages for the transfer of data pertinent to the areas of Radiology, Laboratory and General Practitioners and Hospitals. The laboratory message in particular was studied in depth by the author, and found to contain non-standard elements. These elements can also be found in the other messages designed.

The ramifications of such a message being non-standard are far reaching. One of the more obvious results will be that the message will not be ratified. This section explains the problems with the current message structure and discusses the ramifications of continuing to work with a non-standard message. Finally, ways are suggested by which the message may be restructured to become compliant with the standard.

2.5.1 Overview

The adoption of EDIFACT by the NHS was meant to provide the service with the advantages associated with using such a standard, these advantages can be summarised as:

- Speed;
- Reliability;
- Cost effectiveness.

During the course of this research, a number of projects were underway involving the transfer of medical data, notably the *Good European Health Record* (GEHR) [Gehr95] and *The Sheffield and Hull INterchange of Diabetes Information Group* (SHINDIG) [Grub93], [Dixo99]. It was decided to use the NHS EDIFACT Pathology Message in these projects.

This message was designed by the IMG for the transfer of laboratory data between Pathology Laboratories and General Practice. There is already a comprehensive European message [Exch94] upon which the NHS message is based. The reason for 'tailoring' the European Message was to make it conform to the environment and needs of the UK market. At the time this particular work was being carried out the UK message was at the trial standard stage and had not yet been ratified by the Rapporteur Secretariat.

None of the EDIFACT translator software systems that had been developed at this time in the UK, some of which claimed to transfer medical data using the UK message [Priv94], were used. The reason for this was that a commercial EDIFACT translator [Frat94] not developed for medical messages specifically, was available via the GEHR project [Rouv94]. Quality Assurance (QA) processes had been observed through the design and development of the translator, ensuring compliance with the EDIFACT syntax and structure. Problems first came to light when this EDIFACT translator would not accept parts of the UK message.

2.5.2 The Specific Problems

Three specific areas of concern came to light whilst work was underway.

2.5.2.1 The Segment Tag

As has been shown, the EDIFACT syntax has segments (see section 2.3.1), made up of related simple and composite data elements. Each segment has a tag by which it is recognised in an EDIFACT message. This tag is made up of 3 alpha characters as defined in the United Nations UN/EDIFACT Message Design Guidelines [Sitp92].

The segments defined in the IMG message documentation describe eleven segments that have identification tags that include numeric characters. These are known as segment triggers and take advantage of a proposed 'Snn' trigger segment notation, which allows a trigger segment to start with an S followed by two numeric characters. The European version also uses this notation and declares that:

> *"To overcome [segment collision] problems the implemented message is based upon the so-called 'Snn' solution which is in accordance with the current version of the EDIFACT syntax but requires slight modifications to the existing message design guidelines and rules to be accepted"*[Exch94, p. 90]

This notation was introduced to overcome the problems of segment collision and had not been widely publicised. It was proposed by the *Western European Technical Assessment Group* (WETAG) as a short-term solution [Comm95] in 1994. It was hoped by WETAG, but not guaranteed, that this solution would be incorporated into the next version (version 4) of the EDIFACT syntax. At a subsequent meeting of the Joint Rapporteur Team (JRT) it was rejected [Appendix A].

This creates problems when implementers of the messages using the interim notation try to implement them with converters that do not use this interim notation. The notation has not been formally approved by the UN/EDIFACT Rapporteur as standard EDIFACT syntax or structure. Unfortunately, the position that the IMG have adopted over this has not been made explicitly clear to implementers of the message. Any validated EDIFACT translator will not be able to cope with this, as it is not standard EDIFACT. This renders the message largely unusable.

In a previous version of the *United Nations Trade Data Element Directory* (UNTDED) [Tded93] there is a reference to the service data element '0013', a data element for 'segment tag coded' which allowed up to six alphanumeric characters with the first two characters having to be upper case alpha characters. This service data element has been deleted from newer versions of the UNTDED. Other than this now obsolete reference and the short term WETAG solution there is no provision for segment tags to contain numeric characters.

2.5.2.2 Content of the Segment

'Rule 20' of the Design of UN/EDIFACT Message Guidelines and Rules states:

"A new segment shall not contain the entire contents of an existing segment, nor duplicate the function of an existing segment" [Rule93, p.27]

This promotes a more generic method for the design of segments by ensuring the non-duplication of EDIFACT segments either in part or in full.

In the IMG message documentation for the 'pathology request and report' messages the same eleven segments that use the new 'Snn' notation contain exactly the same simple and composite data elements, in direct contravention of rule 20. Examples of two of these are given below:

SO1	Trigger Segment SG1		
Function:	Trigger segment for segment group number 1 in a message		
No.	Data Element	Status	Rep.
C851	SEGMENT GROUP USAGE DETAILS	C	
9811	Segment group usage, coded	C	an..3
1131	Code list qualifier	C	an..3
3055	Code list responsible agency, coded	C	an..3
9810	Segment group usage	C	an..70

SO2	Trigger Segment SG2		
Function:	Trigger segment for segment group number 2 in a message		
No.	Data Element	Status	Rep.
C851	SEGMENT GROUP USAGE DETAILS	C	
9811	Segment group usage, coded	C	an..3
1131	Code list qualifier	C	an..3
3055	Code list responsible agency, coded	C	an..3
9810	Segment group usage	C	an..70

Table 5

The data element 9811 could be used to distinguish between the contents of the two segments, meaning that one generic segment could be used instead of eleven separate segments with the same data content. The segments as they stand directly contravene 'Rule 20'.

2.5.2.3 "Subset" of the European pathology message

It is claimed that the UK pathology message is a subset of the European pathology message:

> *"The specifications in the NHS trial standard are a subset of the European Pre standard produced by project team PT008"* [Eimg94, p.4]

A subset can be defined as:

> *"a set that forms part of a larger set"* [Cham94]

In mathematical terms it is defined as follows:

> *"If C, D are sets from a universe U, we say that C is a subset of D.....*
>
> *.... if every element of C is likewise an element of D."* [Grim87, p.98]

In fact as stated in the Design of UN/EDIFACT messages Guidelines and Rules:

> *"a sub-set of a UNSM is a message which is directly derived from an approved UNSM, has the same function as the UNSM from which it is derived, and which:*
>
> *i) contains all of the groups and segments defined as having a mandatory status within the message, and the mandatory composite data elements, or data elements within them. There shall be no change of status of the groups or segments contained within the message.....*

....iii) does not add any segments, composite data elements or data elements to the message. "[Rule93, p.36]

The segment tags for segment group 15 for both the European and UK message version are shown below:

European Version:

Segments	*Status*	*Repeats*
SG15	C	99
SPE	M	1
SEQ	C	1
SPC	C	99
PRC	C	9
RFF	C	9
QTY	C	1
DTM	C	1
PAC	C	1
PTY	C	1
FTX	C	1
TDT	C	9
HAN	C	9
LOC	C	9
ADR	C	9

UK Version:

Segments	*Status*	*Repeats*
SG15	C	99
S15	M	1
SEQ	C	1
SPC	M	9
PRC	C	1
RFF	C	1
QTY	C	1
DTM	C	9
FTX	C	9
TDT	C	1
HAN	C	1

In the UK SG15:

1) Segment S15 has been introduced and given the status Mandatory.

2) The segment SPE has been left out even though it has Mandatory status.

3) The segment SPC has changed status from Conditional to Mandatory.

Thus the UK message is not a subset as stated.

The segment group 16 in the report message is similar. The European and UK versions are given below:

European Version:

Segments	*Status*	*Repeats*
SG16	C	99
SPE	M	1
SEQ	C	1
SPC	C	9
PRC	C	9
RFF	C	9
QTY	C	1
DTM	C	99
PAC	C	1
FTX	C	9
TDT	C	9
HAN	C	9
LOC	C	9
ADR	C	9

UK Version:

Segments	*Status*	*Repeats*
SG16	C	99
S16	M	1
SEQ	C	1
SPC	M	9
PRC	C	1
RFF	C	9
QTY	C	1
DTM	C	9
FTX	C	9
TDT	C	1
HAN	C	1

In the UK SG16:

1) Segment S16 has been introduced and been given the status Mandatory.

2) The segment SPE has been left out even though it has Mandatory status.

3) The segment SPC has changed status from Conditional to Mandatory.

Again the UK version is not a subset of the European version.

Similarly for segment group 20 the European and UK Versions are shown below:

European Version:

Segments	*Status*	*Repeats*
SG20	C	99
RNG	M	1
FTX	C	9
CCI	C	9

UK Version:

Segments	*Status*	*Repeats*
SG20	C	9
S20	M	1
RND	C	1
FTX	C	1

In the UK version the segment S20 has been introduced, the segment RNG has been replaced by the segment RND which is defined to do the same thing, but as well as having a changed tag has a changed status.

These examples show that the NHS message is not a subset of the European one either by the dictionary definition of a subset or by the definition given in the UN/EDIFACT guidelines.

2.5.3 The Ramifications of Continuing with a Non-Standard Message

The most obvious result of designing a non-standard message is that it will not be ratified for use by the ratification body. It is clear that the UK message could not currently be ratified for the reasons already given.

Translator software that accepts the message in its present form cannot have been through stringent enough procedures to check compliance with the EDIFACT structure and syntax. This implies the use of non-, or insufficiently, validated software translators that could lead to the production of structurally inaccurate EDIFACT messages. Is it possible, then, to trust the transmitted data to be accurate and reflect the intended meaning? The data may be open to different interpretations by other translators - a direct result of not adhering to the standard.

The main problem when introducing non-standard syntax is that it potentially reintroduces all the accompanying disadvantages of previously used arbitrary forms of passing data.

2.5.4 Possible Solutions

Each problem previously highlighted will be considered separately and the message defined such that it strictly follows the rules of the EDIFACT standard. The results will then be looked at globally. It should be noted, however, that even solving the syntax problems leaves some far more serious basic problems with the use of EDIFACT as a mechanism for transferring clinical data!

2.5.4.1 Solving the Tag Problem

It is accepted that the EDIFACT standard is being constantly redefined and updated and may eventually be redesigned to cope with the 'Snn' form of syntax. However the use of non-standard syntax within a message is a risky procedure. If it is felt that the use of numeric tags is justified in this instance then it should, at the very least, be widely publicised and clearly justified.

2.5.4.2 Solving the Segment Content Problem

The simple answer is to define a single segment, as the contents of the eleven segments are the same. This would also help alleviate the problems of the tag. The segment would have one tag name, for example 'SOG', with the content not being repeated elsewhere and importantly would not contravene the design rules.

However, the reason that the 'Snn' notation was adopted in the first place was to avoid the problems caused when segments are used as trigger segments in a complex message. Trigger segments are segments that appear first in a segment group and are mandatory. They help to maintain the logical flow through a message implicitly by virtue of their position and are used by the translator software. Many problems can be caused if the translator cannot maintain its position within the message, one such being segment collision [Tded93]. Segment collision can be initiated if identical segment triggers appear in adjacent segment groups. This would be the case if the eleven segments were merged into one with a single tag name.

A solution to both problems is to delete the original segments from the message and ensure that the new segment triggers are not identical to adjacent ones in the segment groups. However, although this would conform to the EDIFACT standard, it is not satisfactory due to the difficulties in predicting whether adjacent segments are the same in complex messages. The long-term solution may be to adopt a method similar to the 'Snn' notation but this has to be part of the accepted standard before message designers implement it in messages.

Once again, clear documentation and justification of the chosen solution is vital in order to retain confidence in the message.

2.5.4.3 Solving the Subset of the European Message Problem

The obvious solution is to state that the UK message is *based* on the European one and is not *a subset*. The alternative is to delete the extra segments. If the additional segments are needed in the UK, the latter solution cannot be adopted. However, this should not be dismissed without due consideration. Bearing in mind that much research and development work went into the building of the European message, users of the message could have greater confidence in it if the reasons behind the changes were clearly explained and justified.

2.5.4.4 Overall

It is generally recognised that a better method for structuring trigger segments within the message is needed and that the introduction of the 'Snn' notation is a step in this direction. The trouble lies in using it before the appropriate rules and guidelines have been changed to accommodate it and before it has become part of the EDIFACT standard. One of the problems is simply that this use of non-standard syntax has not been clearly explained. The issue has been 'avoided' such that many implementers are not even aware of it. Without full explanation and justification it is hard to have confidence in the message as a whole.

2.5.5 Conclusion

A clear explanation of how the IMG arrived at the structure is needed in order that implementers and users of the message can have confidence in it. One would naturally expect the IMG to adhere to the rules, guidelines and standard syntax for the design and implementation of a UN/EDIFACT message. If they do not, justification is essential.

Some criticism could be levelled at the UN/EDIFACT structure and syntax design. Consider the maximum number of segments allowed in use at any one time. There is a theoretical limit of $26^3 = 17576$ segment tags, this being the number of characters in the alphabet to the power of the length of the segment tag, to cover all possible segments needed by every organisation that use EDIFACT. The tag is meant to give some indication of the intentional use of the segment, the tag being used as a mnemonic. It appears that the original designers of the EDIFACT syntax underestimated possible future needs. However, the syntax and structure are under constant review and development in order to tackle this type of problem.

If flaws are found in the basic EDIFACT syntax, then these must be tackled but at the correct level and in the right way, i.e. by making representation to the Rapporteur Advisory and Support Teams in conjunction with the UN/ECE Secretariat, who must then take action to alleviate problems. It goes against the whole philosophy of the use of standards to have user-defined solutions [Comm94] being introduced in an ad hoc way.

Disseminating a 'standard' that is *non-standard* could lose the very real advantages that standards have already brought, and will continue to bring to the NHS.

2.6 GEHR Object Model in EDIFACT

It has been shown in the previous sections that difficulties will accrue from the use of the NHS designed EDIFACT messages that suffer from the problems illustrated. However, there is a definite need for the EDI of medical data. This creates a problem and this section highlights the research that was undertaken to find a way of transferring data conformant to the GEHR architecture (see section 4.4) using the EDIFACT syntax.

2.6.1 Overview

When transferring data between different sites it is essential that it is done in a structured manner and in a way in which both the sending and receiving sites understand. Importantly, it should be done in a way that provides an adequate level of security and confidentiality [Ross95].

One result of the GEHR project was the design of an architecture [Gehr95] for the standard recording of patient data. Because the ideas from the GEHR project were being fed in to the standards making bodies such as CEN, it was thought that a useful exercise would be to create a *GEHR Object Model* (GOM), Appendix B, message using the adopted NHS standard EDIFACT syntax for the transfer of data structured using the GOM

This section highlights the design decisions that have been made when designing an EDIFACT message that is compatible with the GOM. In general, each class in the GOM was translated into an EDIFACT segment and the attributes in the classes were translated into EDIFACT data elements.

After attempting to design a GOM EDIFACT message, EDIFACT was shown to be inadequate for the purpose. While the GOM provides for comprehensiveness and full flexibility in the clinical context, EDIFACT, initially designed for messages with an administration, commerce and transport bias is inadequate for the transfer of comprehensive medical data.

2.6.2 EDIFACT Design

Each cluster, defined in the GOM [Gehr95], was taken in turn and EDIFACT syntax designed for it.

2.6.2.1 EHCR Cluster

The data that would appear in the Extract segment was not finalised but it was certain that it would contain the following data items:

Date/Time – the extract was sent

Transaction List – a list of the transactions sent

Health Care Professional (HCP) – the clinician responsible for the sending of the extract

Health Care Facility (HCF) – the institution the message has been sent from

2.6.2.2 Transaction Cluster

2.6.2.2.1 Acquired_Versioned_Trans

When data is being transferred to a new site the attributes of *Acquired_Versioned_Trans* (AVT) are sent with it. This class inherits from *Versioned_Trans* (VT). These two classes when aggregated may be defined as a single segment. The attributes can be modelled as EDIFACT data elements.

In order to represent the GOM, which allows many repeats of *Trans_Version* (TV) (i.e. allows multiple *Admin, Summary, Report, Cont_Care, Nota_Bene, Contact* and *Trigger* transactions), a segment group should be initiated by a VT for each separate transaction (see Figure 5). The class TV is deferred which means that the attributes that it contains will be shown in the segment of the first concrete class that inherits from it (a particular transaction type). The same is true for *Standard_Trans*.

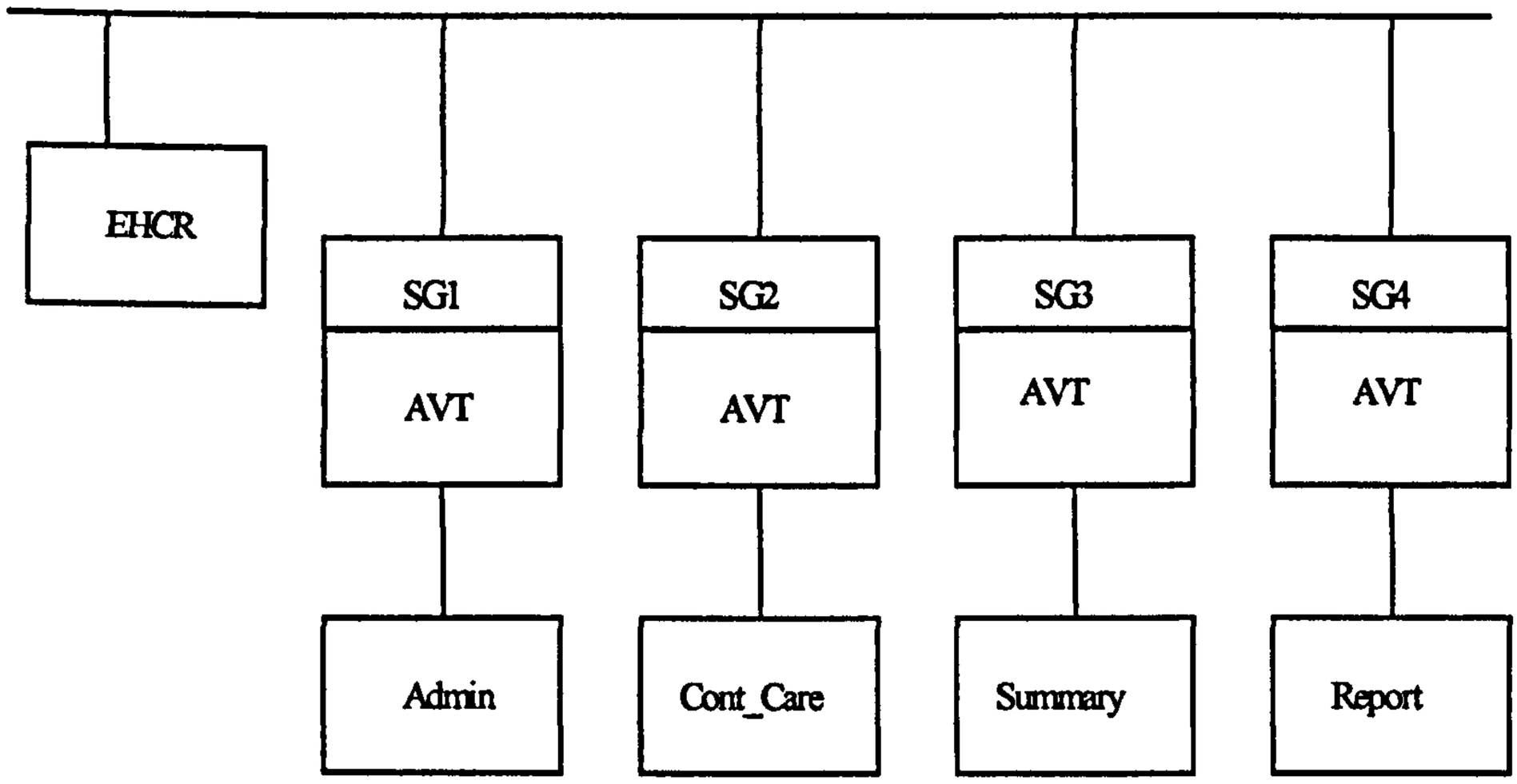


Figure 5

Unfortunately the diagram modelled in Figure 5 will suffer from a problem known as segment collision. This occurs when consecutive segment groups are triggered by the same segment. The consequences of this are that the message will become unprocessable rendering it useless. (see Section 2.5.4.2)

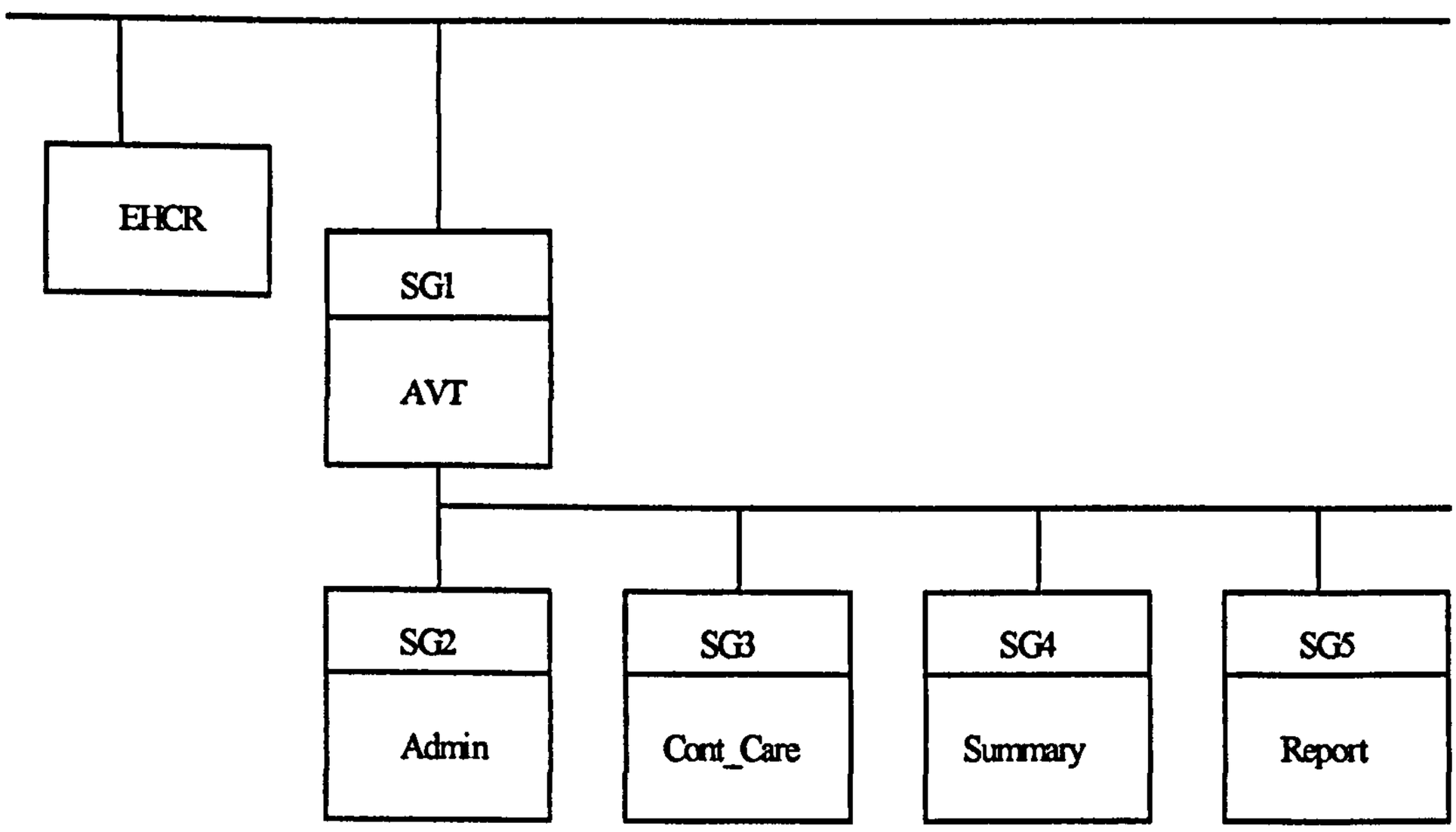


Figure 6

A solution that avoids segment collision is shown in Figure 6. Different transactions can occur many times after the segment AVT. This is allowed in the architecture, although not in the spirit of the GEHR philosophy [GEHR95, section 5.3]. The work that was carried out at the time highlighted many issues like this that were valuable lessons. These lessons fed directly back into follow on GEHR projects and eventually

into the CEN standards for healthcare architectures. Whilst the work presented here was valuable in highlighting shortcomings, work in the area of the architecture has moved on and is shown later in the thesis.

The decision was taken to continue with this form of the message as it avoided segment collision. This decision was also based on the assumption that any GEHR compliant system would not allow transaction versions that contained different transaction types. Ensuring the message that transferred the data to another site, although theoretically able, would not contain transaction versions of different types.

The AVT segment is allowed to have up to 9999 repetitions – the maximum allowed by the EDIFACT syntax.

AVT will include the following data items:

ID
Date/Time
Access Rights
Amend Rights
GEHR Version
HCP ID
EHCR Source Pointer
Source Transaction Reference
Was Gehr Source

2.6.2.2.2 Transactions

Each transaction is explicitly defined in the message taking all the attributes of the deferred class Trans_Version and then adding their specific attributes as data elements.

There are to be six transactions represented in the proposed GOM EDIFACT message, Trigger being left out at present as it plays a slightly different role from the other transactions within the record. Because of the problems with segment collision, each one of the six transactions starts a segment group (see Figure 6). Since they are defined as different segments, the problem of segment collision is avoided. The data elements that all six transaction segments have in common are:

Revision
Date/Time
HCP
Change_Type
Recorder

Then for each transaction:

Transaction Name	**Extra Attribute**
Summary	Date_Range
Report	Observation
Cont_Care	Date_Range
Admin	Patient
Contact	Date_Time

The Nota_Bene requires no additional attributes.

A problem occurs here when defining segments that have different names but contain the same data elements. They can be seen to be doing the same job. This is illegal in the EDIFACT syntax [Rule93]. Taking Cont_Care and Summary as an example, they both contain all the attributes of Trans_Version and add their own, which happens to be the same in this case, Date_Range. If this is to be the case then the segment should be merged and a qualifier used to distinguish in which transaction type the segment is to be used at any instance. However, this would not cater for any future changes that may take place, such as additions of different attributes to each transaction class. It would also be inconsistent with the way in which the other transactions are designed. As the flexibility that is embodied in the GEHR architecture is essential for the portability and communicability of the data held in the structure, any syntax used for the transfer of data in this format that displays inadequacies is unsuitable. This highlights that it is inappropriate to use EDIFACT as syntax for the passing of transaction information.

2.6.2.3 Item Cluster

2.6.2.3.1 HRIs and Collections

It was thought that the *Health Record Items* (HRIs) and *Collections* may be put in a single segment with a qualifier stating the type of each segment occurrence. However, this was found to be unsuitable as Collections have the extra attribute *members*, which is a list of other HRIs and Collections. Also, the HRIs are made up of many other attributes that have no place within Collection. This would have the effect of leaving many data elements null when using the segment as a collection.

It was at this point that other problems appeared. Collections are naturally recursive: it is very difficult to model recursion in EDIFACT, as a segment within a segment or a composite data element within a composite data element is not allowed. The data

element members cause the recursion, which is a list of Collections and HRIs that in turn can also have multiple members. In order to model this, identification would have to be given which would point to the occurrence of the Collection or HRI referred to within the Collection.

To facilitate the pointer, already described, an Identification (Id) segment was introduced which contains the Id number and a qualifier. The qualifier is needed as the segment could be used for different Ids such as GEHR_UID or OBS_ID. This segment is to follow the Collection or HRI in order to identify the Id of that Collection or HRI (see Figure 7). Both the Collection and HRIs can be repeated up to 9999 times. This is a limit that has been imposed on the design by the EDIFACT syntax. In reality there should be no limit, however it is not envisaged that the number of Collections and HRIs will exceed 9999 in one transaction.

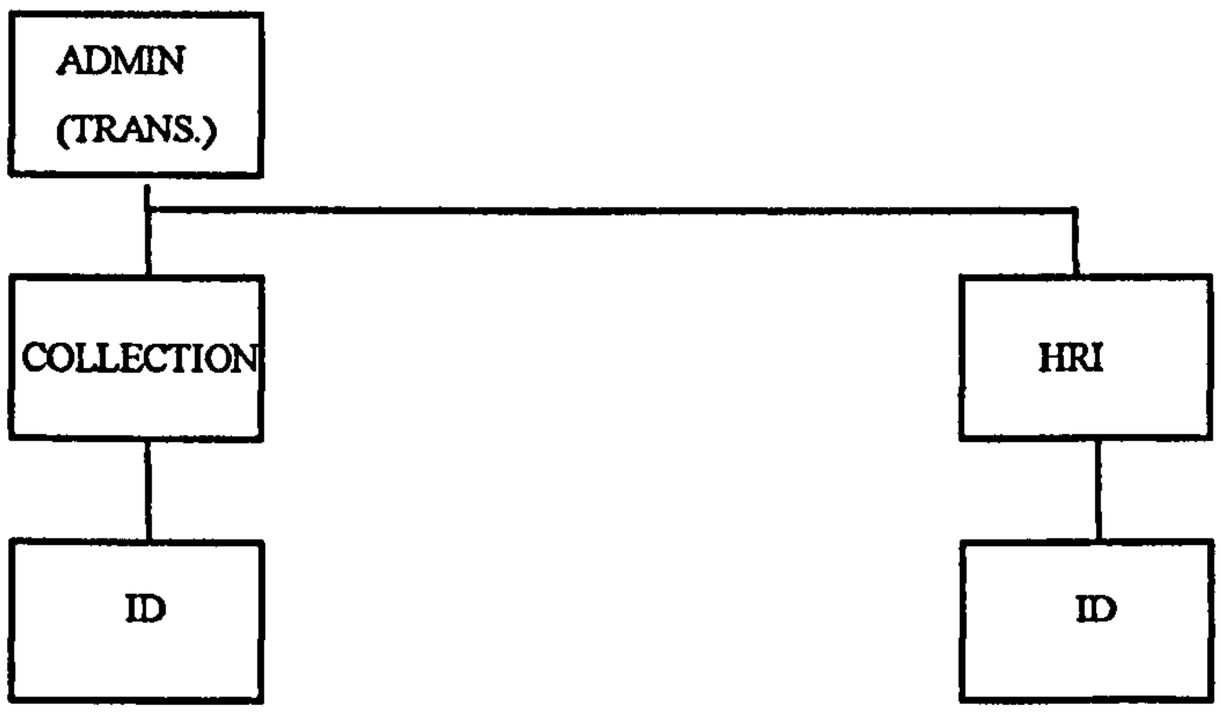


Figure 7

2.6.2.3.2 Observation

The information held in the observation class:

Info_prov
Access_Rights

Also, EHCR_Entry data will be held here:

Name
Emphasis
Recorder
Shadow_auth

These will all be data elements in an observation segment. The problem of recursion again arises at the observation level, as the attribute in_reply_to, references an

observation. This is dealt with in the same way as Collections and HRIs by having an ID segment following the observation to specifically identify which observation it is.

2.6.2.3.3 Heading

The information held in the heading segment will also be the EHCR_Entry composite and parent which is a data item giving the parent of the heading.

2.6.2.4 Quantity Cluster

It was decided to make the quantity cluster a segment which was itself made up of many composites. These composites being the classes found in the quantity cluster. When these composites had been modelled, other composites that did not have any attributes were modelled. This led to having composites within composites is illegal in the EDIFACT syntax.

2.6.2.5 The EDIFACT Structure

The EDIFACT structur at this point can be seen in Figure 8.

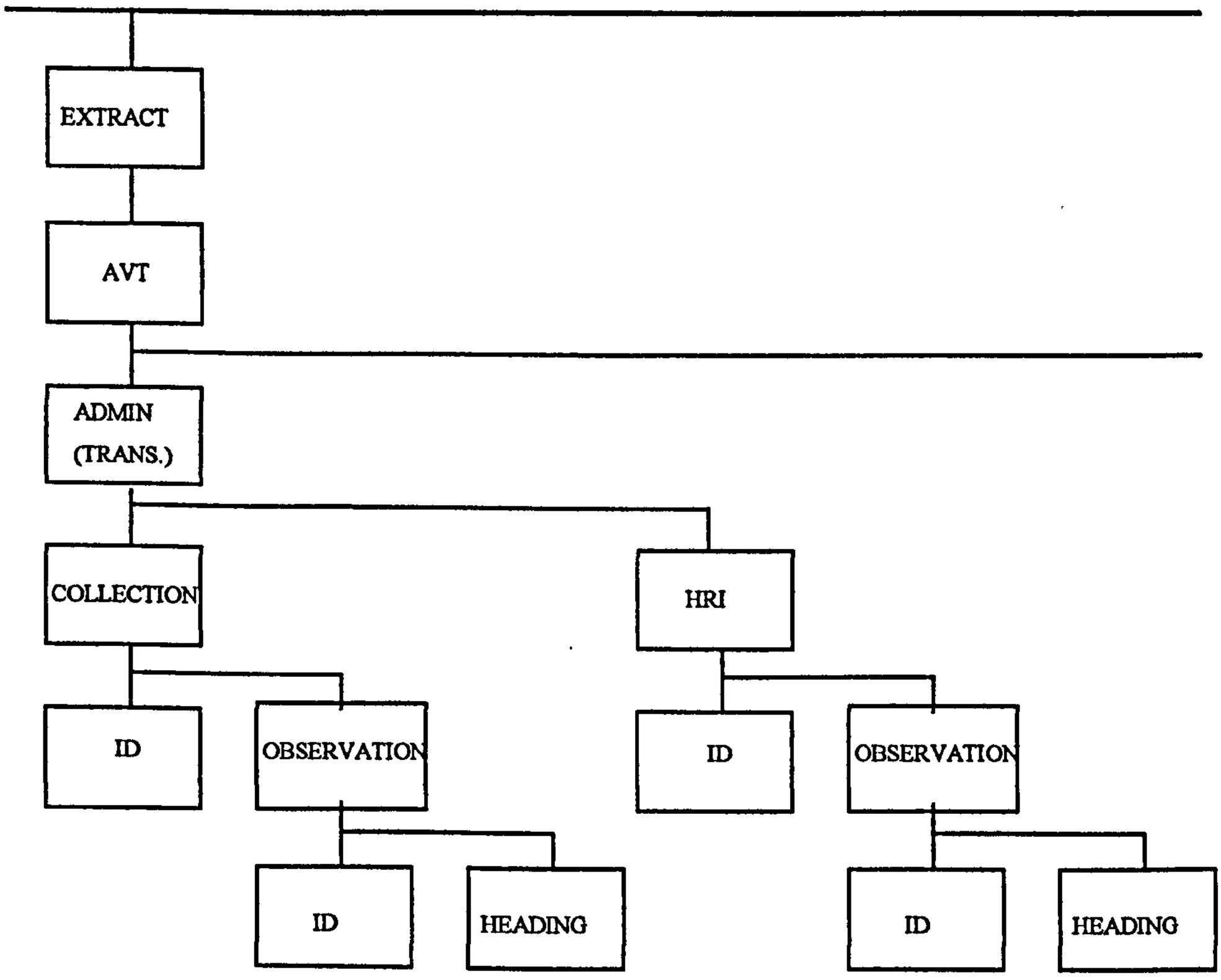


Figure 8

2.6.2.6 Design Decisions

The problems experienced when trying to design a GOM message led to the decision that EDIFACT is unsuitable to be used for the exchange of GEHR data, for the following reasons:

- Segment Collision could easily occur.
- Recursion cannot be satisfactorily modelled.
- Segments/Composites cannot be designed with other Segments/Composites within them that is needed to facilitate the GOM structure.
- The GOM cannot be modelled using the EDIFACT syntax.
- The resulting message will be convoluted with lots of pointers to other segments.
- Segments are frequently repeated throughout the resulting message.
- The lengths of data items in EDIFACT have to be explicit, which can not be predetermined in the GOM. Which was designed specifically to allow for flexibility when recording data.
- EDIFACT can not handle the transfer of data items such as video and image material, or *Bulky Data* as it is known in the GOM.
- EDIFACT is linear whereas the GOM is not.

2.6.3 Alternative

It has been shown that the whole of the GOM cannot be represented using EDIFACT so an alternative solution has to be found. An EDIFACT message could be designed to hold the information important to that of an Extract. This will be information that is important to know so that anything or anyone receiving the message knows what to do with it. This proposed EDIFACT message would effectively act as a header to an alternative non-EDIFACT message form that would contain all the relevant information about the patient(s).

This method would also be in keeping with the NHS policy of transferring data using an EDIFACT message, as it would arrive at a site with an EDIFACT header, which

could be translated to show the format of the remaining information. In this way no healthcare site should be in the position where they receive a GEHR related message they do not know what to do with.

2.7 Proposed Message for Transfer of Data

2.7.1 Introduction

It has been shown that the EDIFACT syntax is wholly inadequate for the exchange of GEHR data. Also it has been shown that the EDIFACT syntax is inappropriate because of some of the design issues that have been taken whilst developing EDIFACT. It is for these reasons that another method for the transfer of data had to be devised.

2.7.2 Design of EDIFACT Extract Header

The information being transferred in a header will not be affected by the shortcomings of the EDIFACT syntax. If a message were received at an NHS site it would be understood because EDIFACT is the standard adopted by the NHS, for the passing of health related messages. This method promotes the automatic handling of messages from multiple sources.

The information that needs to be transferred in the header is:

- Date and time of the creation of the extract.
- HCP information - to show who is responsible for the creation of the extract and where the extract originated.
- The EHCR source - to show the electronic source of the information.
- GEHR version - to show the version number of the GEHR extract.
- Information to indicate that what format the attached data is in.

2.7.3 Proposed Header

The proposed header message can be seen in Figure 9. The UNH, BGM and UNT segments are standard EDIFACT segments their function is explained below:

UNH - To head, specify and identify a message.

BGM - To identify the function of a message and to transmit the identifying number.

UNT - To end and check the completeness of a message.

The EXT is an additional segment, designed specifically to hold the details identified in the GEHR Extract class, details of which are given below.

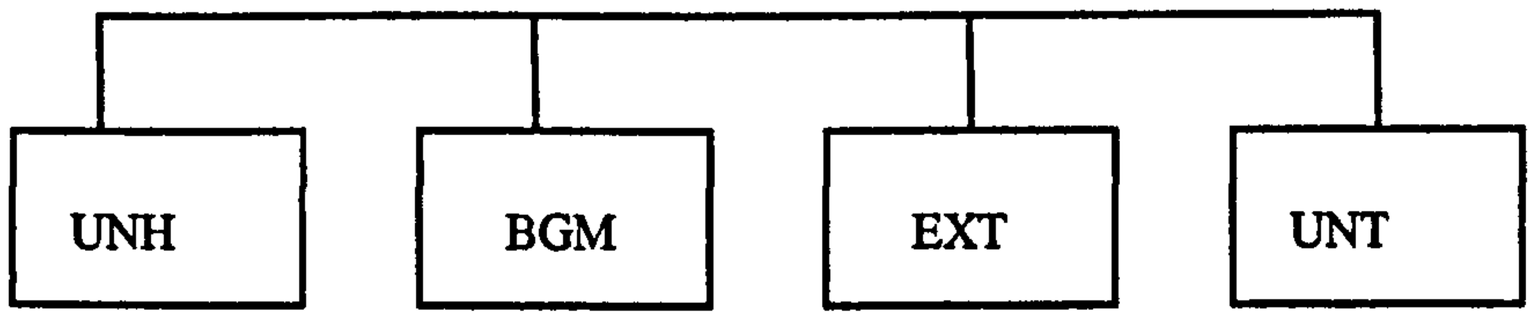


Figure 9

The Extract segment designed to contain specific relevant information can be seen in Table 6.

EXT - EXTRACT Function: To specify information about the Extract to be sent				***Explanation of the elements in the Extract:***
7402	IDENTITY NUMBER	M	an..17	***EHCR_UID***
4440	FREE TEXT	M	an..70	***Name of EHCR Source***
4440	FREE TEXT	C	an..70	***Net Address of EHCR Source***
C058 3124 3124 3124 3124	NAME AND ADDRESS Name and Address Line Name and Address Line Name and Address Line Name and Address Line	C M C C C	 an..70 an..70 an..70 an..70	***Name and Address of HCF***
3251	POSTCODE ID	C	an..9	***Postcode of HCF***
C076 3148 3155	COMMUNICATION CONTACT Communication Number Communication Channel Qualifier	C M M	 an..25 an..3	***Contact Number of HCF***
C076 3148 3155	COMMUNICATION CONTACT Communication Number Communication Channel Qualifier	C M M	 an..25 an..3	*" "*

4440	Free Text (Net Address)	C	an..70	***Net Address of HCF***
C507	DATE/TIME/PERIOD	C		***Date and Time of the creation of the Extract***
2005	Date/Time/Period Qualifier	M	an..3	
2380	Date/Time/Period	C	an..35	
2379	Date/Time/Period Format Qualifier	C	an..3	
C058	NAME AND ADDRESS	C		***Name and Address of HCF Creating Extract***
3124	Name and Address Line	M	an..70	
3124	Name and Address Line	C	an..70	
3124	Name and Address Line	C	an..70	
3124	Name and Address Line	C	an..70	
3251	POSTCODE ID	C	an..9	***Postcode of HCP***
C507	DATE/TIME/PERIOD	C		***Date and Time HCP Address Valid From***
2005	Date/Time/Period Qualifier	M	an..3	
2380	Date/Time/Period	C	an..35	
2379	Date/Time/Period Format Qualifier	C	an..3	
C076	COMMUNICATION CONTACT	C		***Contact Number of HCP***
3148	Communication Number	M	an..25	
3155	Communication Channel Qualifier	M	an..3	
C076	COMMUNICATION CONTACT	C		*" "*
3148	Communication Number	M	an..25	
3155	Communication Channel Qualifier	M	an..3	
4440	FREE TEXT	C	an..70	***Net Address of HCP***
4440	FREE TEXT	C	an..70	***Grade of HCP***
C846	SERVICE PROVIDER POSITION	C		***Position Details of HCP***
3813	DETAILS	M	an..3	
1131	Service Provider Position, Coded	C	an..8	
3055	Code List Qualifier	C	an..3	
3812	Code List Resp. Agency, Coded	C	an..35	
	Service Provider Position			
C844	SERVICE PROVIDER TYPE	C		***Profession Details of HCP***
3829	DETAILS	M	an..8	
1131	Service Provider Type Identification	C		
3055		C	an..8	
3828	Code List Qualifier	C	an..3	
	Code List Resp. Agency, Coded		an..35	
	Service Provider Type			

3206	COUNTRY	M	an..17	***Country of Reg of HCP***
7402	IDENTITY NUMBER	M	an..17	***Reg Number of HCP***
4000	REFERENCE VERSION NUMBER	M	an..35	***Ref to GEHR Version***
C999	ATTACHMENT	M		***Attachment type***
9800	Attachment type	M	an..70	
1131	Code List Qualifier	C	an..8	
3055	Code List Resp. Agency, Coded	C	an..3	

Table 6

2.8 Summary

In this chapter EDIFACT, as adopted by the NHS for the transfer of health data has been explored. The messages subsequently designed to transfer this data, one of which was based on the European message that had previously been produced, was investigated in detail.

The study of this message showed that it was flawed in several areas. The errors that were highlighted in this chapter also manifested themselves in other messages that the NHS IMG has developed.

The GEHR architecture that is feeding into the emerging European standard for the storage of medical data in an electronic healthcare record was considered. It was thought to be a useful experiment to put GEHR data in an EDIFACT message. This was shown not to work. As a result of this a method to transfer data in a way consistent with the NHS guidelines for the transfer of health data was needed. This has been facilitated by designing an EDIFACT header that could be attached to the front of a byte stream of data giving details of the format of the byte stream as well as important information about who sent the message, dates and times.

The remainder of this thesis discusses the investigations that were carried out to design and use an appropriate syntax to be appended to this EDIFACT header.

Chapter 3

3 Alternative Methods of Data Transfer

3.1 Introduction

Having studied EDIFACT and concluded that it is not feasible for the transfer of medical data, other potential methods for transfer of medical data were investigated. This chapter studies these methods and analyses the usefulness or otherwise of each. In particular Health Level 7 (HL7) version 2, Accredited Standards Committee (ASC) X12, Synapses, Synex, Extensible Markup Language (XML), the work of CEN TC/251 WG1 and CORBA™ are considered.

3.2 Health Level 7

3.2.1 Introduction to HL7

HL7 is a protocol developed for the electronic interchange of clinical, financial and administrative information among independent healthcare computer systems.

HL7 as an organisation was founded in 1987 at a conference at the hospital of the University of Pennsylvania, since when its membership has grown from 300 to in excess of 1,500 members. Made up of healthcare providers, vendors (who are often competing) and consultants. The participants share a common goal of simplifying the implementation of interfaces between computer applications from different vendors. They aim to standardise the format and protocol for the exchange of certain key sets of data among healthcare computer application systems.

3.2.2 Background

ISO 7-layer reference model for OSI

The term "level 7", in the name Health Level 7, refers to the highest level of the *Open Systems Interconnection* (OSI) reference model of *the International Standards Organisation* (ISO).

The highest level is the application layer of the standard. Things that are of concern at this level include [Mars96]:

Identification of the intended communication partners

Establishment of the necessary authority to communicate using the OSI environment

Determination of the availability of the intended communication partners

Agreement of privacy mechanisms as required for the communication

Authentication of the intended communication partners

Determination of allocation of the cost of using the necessary resources

Determination of the adequacy of the resources available for the intended communication

Synchronisation between co-operating applications

Agreement of who has responsibility for error recovery

The reference model was developed during the mid 1970's and was completed in 1979 when it became the ISO 7498.

Each of the seven layers in the model contributes to the sending and receiving of data in an open systems environment. Data is surrounded by extra pieces of information relevant to each layer as it passes through each level from 7 to 1. When the recipient has received the package of data and other information it passes through level 1 to 7. At each level the appropriate information is unwrapped until it reaches the application layer. If the transfer has been successful (error messages are produced if not) the data that was passed can then be viewed by the user of the application e.g. the application could be a database with the user issuing a query or similar.

3.2.3 How HL7 Relates to the 7 Layer Model

It is stated in section 1.2 of the HL7 specification that level 7 refers to the ISO 7 layer reference model. However, it goes on to say that it *does not conform to the ISO defined elements of the OSI's seventh level.* Also, it says that it *does not specify a set of ISO approved specifications to occupy layers 1 to 6 under HL7's abstract message specification.* The relationship between HL7 and ISO 7-layer model for OSI is conceptual. It is meant to conform to the definition of an application-to-application interface, taking into account some of the matters above.

The real relationship would seem to be in name only.

3.2.4 Critique of HL7

Although HL7 has become well established over the last ten years it makes several assumptions which the following sections argue are fundamentally incorrect. This section highlights some of the inadequacies with the assumptions set out in Version 2.3 of the HL7 specification.

It is stated that the standard is to be used for the transfer of "certain key sets of data" rather than all data pertaining to healthcare, which would be more desirable. The specification also states that it is hoped that the use of HL7 will "yield a voluntary *ad hoc* standard" this is surely something that should be avoided at all costs. If an international standard is not forthcoming then the very least that should be expected is an industry *de facto* standard rather than an *ad hoc* one which surely will only lead to chaos.

As is shown in the specification document [HL7-96] the HL7 specification is well established with over 300 members contributing to the standard at each quarterly meeting. The standard is still evolving. In order to be an HL7 user a healthcare institution has to either purchase the standard or use it through a member vendor.

HL7 exhibits many of the inadequacies that are inherent in other standards (such as EDIFACT). One of these is the use of a 'Z segment' for the passing of information that is site specific. The problems associated with this are:

Each site is free to make up its own site-specific message segments. This is adequate if the data is only being passed internally but if the data is needed outside the institution then HL7 fails in this respect as the passing of non-standard messages would take place.

Site-specific codes can be used, this is adequate if you can also pass all the information that is associated with this code e.g. who is responsible for it, who maintains it, where it originated, etc. However, there is no scope for finding out these details and if another institution were to receive a message containing them without this extra information the code would be inadequate.

The same is true of maintaining site-specific tables

HL7 maintains that because "*of the diverse business process that exist within the healthcare delivery system...the healthcare delivery system prevents the development of either a universal process or data model to support a definition of HL7's target environment. In addition HL7 does not make a-priori assumptions about the architecture of healthcare information systems nor does it attempt to resolve architectural differences between healthcare information systems*"

This statement implies many things:

Firstly, that in order to transfer data in a standard way an architecture for healthcare data is needed.

Secondly, without an architecture the passing of medical data cannot be carried out properly

Thirdly, it explicitly states that for the reasons given HL7 is not a true 'plug and play' interface standard. This means that one healthcare provider cannot receive or send any data to another party without having prior negotiations about what it is they are to pass between them. This is not the best situation to be in, having to agree what data will be passed between different institutions before any data can be sent. Even though agreed data sets may be adequate the majority of the time, it is a fact of the healthcare environment that situations will occur were data containing unusual items will need to be communicated quickly, without resorting to lengthy processes of agreement on formats.

As with other transfer standards HL7 provides an electronic data dictionary of all data elements that can be used. Again problems are encountered due to this standard not being flexible enough. If a new laboratory test was carried out and a pathologist wanted to transfer this information, they would have to wait for this to be put into the data dictionary so that everybody knew about it. The alternative would be to pass site-specific data elements but this is inadequate, as agreement on the meaning of these data elements would have to be settled before any data could be transferred.

Data fields are found in the message by virtue of the position of their associated segment. This means that only known sets of data can be transferred.

HL7 distinguishes between fields that have the null value and those that are not present. The former is represented by two adjacent quotation marks, the latter by no data at all.

This sentence highlights the problems that are encountered when the data is not separated from the syntax that is used to transfer it. It also highlights the need for an underlying model or architecture but as has already been pointed out HL7 makes no provision for either. It in fact goes completely the opposite way by explicitly saying that it does not intend to adhere to any data model.

The encoding rules for HL7 state *"a receiving application should ignore fields that are present in the message that are not expected rather than treat such a*

circumstance as an error". Firstly the recipient of a message should be able to read any data that has been sent to them that they were not expecting. The transfer mechanism should cater for this. However if it does not it should surely be throwing up an exception rather than ignoring the data all together. Consider the scenario were x relies on y to make clinical sense and both are sent. The recipient expects x therefore y is ignored. The result of this would mean that x could have a totally different interpretation. Integrity of the data is compromised.

The specification document states *"The HL7 standard is intended to standardise data interchanges, not the underlying application systems."* In order to achieve fully flexible data interchange the data that is transferred should be based on an information model or architecture. If an architecture is suitable for the exchange of data then it could be argued that the same architecture could be used for the storage of data at source. Without an architecture only agreed data will be passed in an *ad-hoc* fashion. In order for the resultant method of data transfer to be comprehensive the issue of data transfer has to be viewed not just from the point of view of the transfer itself but from the wider perspective of storage and manipulation of the data.

The specification says that all standards must evolve as the applications they support change and also due to the result of experience using them. This is very true, the experience of using HL7 shows that it is limited in the way it transfers data. It was originally intended for transferring data around only single hospital sites all using the same type of mechanism for the storage of data. It has evolved to being used for the transfer of data between many different types of healthcare providers and also across many different areas of medicine. Because HL7 is being used in a very different way than it was first designed for it limits the type of data that can be transferred.

As with EDIFACT, the HL7 consortium has defined a list of codes that can be used for the transfer of data in a seamless way. This is a good step to take but is not really the remit of a group tackling the subject of data transfer. As it is in the remit of coding agencies.

HL7 has given a lot of thought to the request result cycle and seems, in this respect to go deeper than do other exchange mechanisms such as EDIFACT. As well as data transfer, q*ueries* can also be made from an application system not necessarily holding

information about a particular patient to another application system that does hold the information needed.

HL7 has also given thought to other communications environments other than the ISO 7 layer model. It has produced basic assumptions utilised by HL7 when communicating for these other environments if they wish to transfer medical information using HL7.

The specification states that it does not care how individual systems actually store data within an application once it has arrived. This is a valid statement as to do so would be going outside the remit of data transfer mechanisms. However it then goes on to describe methods for updating a record when receiving a null value or when an optional field is omitted. As stated this is outside the remit for HL7. However, it is worth noting how these two scenarios are handled:

When receiving a field with an optional value omitted the application should not update the record in a database but should leave the old value unchanged.

When receiving a field with a null value, represented by two quotation marks (“”), the record in the database should be changed to null.

This has serious connotations, it is suggesting the retrospective changing of data by overwriting a previous record already held in the patient database which has associated with it a whole host of problems such as medico/legal implications.

HL7 can handle the transfer of any graphics file that conforms to a *Multipurpose Internet Mail Extensions* (MIME) format. It does this by use of its *Encapsulated Data* segment. It can also support various waveform data, in this respect it can encompass a wider type of data than EDIFACT.

3.2.5 Conclusions

Whilst it has been shown that there are many anomalies with version 2 of HL7, most recently the HL7 group have dedicated their work to the release of Version 3 of HL7. This is radically different from all previous versions of HL7. It introduces a Reference Information Model (RIM) that provides an information model or architecture of the messages. There are some key concepts in version 3 that differ from previous versions. Version 3 specifies a means of identifying the responsibilities of the senders

and receivers of messages. It also identifies a common description of the exact fields of a message and their grouping, sequence, optionality and cardinality.

It would seem that although previous versions of HL7 held some inherent problems version 3 represents a change in ideology which will influence the way in which messaging is carried out.

3.3 ANSI ASC X12

The American National Standards Institute (ANSI) in 1979 set up the Accredited Standard Committee (ASC) X12 for the development of standards for the inter-industry electronic interchange of business transactions. The main purpose of ASC X12 is to develop, interpret, publish and promote the proper use of American National and UN/EDIFACT standards.

ASC X12 promotes the exchange of data in all areas and is not just specific to healthcare data. Other areas they cover include Education, Finance, Transportation and the Insurance industries. Due to the disappointing take up of the EDI standards adopted by ASC X12 as admitted by ASC X12:

> *"EDI implementation has not reached the level that was long expected"* [Feat98]

ASC X12 has sought to change direction and has started considering the next generation of EDI standards. As part of this analysis ASC X12 realised that *"the use of object oriented architectures permit applications acquired from different sources and installed on different platforms to freely exchange information"* [Feat98]. The next generation of standards proposed by ASC X12 should be based on an underlying model or architecture of the business area. As ASC X12 put it, the next generation of EDI standards they will produce will be *"virtually a complete makeover of the standards body and development process."*

In summary, it has now been formally recognised that the standards that have been adopted by ASC X12 are not the way to go, but that the next step includes basing the next generation of standards on an object model or architecture of the business process to achieve what is known as OO-EDI.

3.4 Synapses

3.4.1 Synapses Overview

Synapses was a European funded project under the umbrella of the Health Telematics R&TD framework, which concluded in December 1998. The aim of the project was to allow for shared care of patients by enabling healthcare professionals to access patient record information from distributed and diverse healthcare information systems.

The Synapses view of the world is that information about an individual patient is distributed across both primary and secondary healthcare, this information is also stored on different systems within each establishment. Some of the data is stored in legacy systems. This is different from the view of information being held about a patient in a central repository. The individual systems on which the information is held are known as specialist feeder systems.

In order to share information between these diverse systems the idea of a middleware server has been introduced. These servers take a request for information from a clinical workstation and use a pre-defined dictionary to elicit the data in a standard format from the feeder systems and then forward the response to the clinical workstations.

The Synapses approach utilises the methodology of the database federation to a standard and comprehensive schema, the federated healthcare record architecture, mediated and managed through a set of middleware services.

3.4.2 The Federated Healthcare Record

The Synapses server holds a virtual record of a patient. Each of the feeder systems is interrogated to determine what information each holds about the patient that is being investigated. The server initiates formal object requests for record extracts from each of the feeder systems. The object requests are made in the form of Synapses Objects. The Synapses Federated Healthcare Record (FHCR) architecture is based on the European pre standard architecture preENV 12265.

The information held on the feeder systems is interrogated by the server using a pre-defined object dictionary, the Synapses Object Dictionary (SynOD).

3.4.3 The Synapses Object Dictionary

In order for the server to communicate with the feeder systems the data that is being interrogated in each of the systems must conform to the same types. The object dictionary that has been outlined by the Synapses project defines the complete set of object templates that will be held across the federation of feeder systems. The dictionary provides a means by which the server can elicit information from the client.

3.4.4 Considerations

Synapses recognises the need for a standard architecture before any communication can be carried out between heterogeneous systems. It has therefore used the preENV 12265 architecture. However it is concentrating on communicating between legacy systems rather than looking to the future.

The Synapses project having utilised the preENV 12265 architecture has had to build upon this at the Record Item Complex and Record Item level, in order to define several specialised sub-classes with specific roles within the FHCR. This shows that the preStandard is not comprehensive enough for actual use.

By defining data types for each specific clinical concept the SynOD is too prescriptive and too strict.

It is thought that in time the SynOD will encourage data to take the same particular data types on diverse systems. Again this seems too prescriptive and impinges on the right of the clinician to enter information in any way that they see fit [GEHR92]

3.5 SynEx

A follow on project from Synapses is Synex. This project has been given the brief to address the issues inherent in the provision and use of multimedia patient records across large enterprise-wide networks. It will extend the work on the architecture of the distributed EHCR beyond Synapses by incorporating the terminological work of the Galen-in-use [Rodr97] project. It will also endeavour to provide middleware components to facilitate the sharing of EHCRs across open distributed computing platforms.

The brief it has been given tends to suggest that the Synapses project has not completed the work on sharing data and communicating data between heterogeneous systems.

3.6 XML

XML – Extensible Mark-up Language – is the proposed successor to the Hyper Text Markup Language (HTML) on which Web pages are based. It uses the word *Extensible* as it allows those providing documents to define their own, new tags in a standard manner. This brings in greater flexibility when viewing documents over the web. It provides pages with greater interaction than the standard HTML. XML is based on the Standardised Generalised Markup Language (SGML) which was designed as a low-level tool-kit to enable the development of customised text processing systems.

SGML was used in the publishing to select or mark up various features that were felt to be important for subsequent processing of the document. For instance a particular word may be tagged so that it could be retrieved at a later date. When the document was then presented to the reader parts of the document would be displayed in a different typeface or italicised. It was a way of conveying meaning to the text by the author.

3.6.1 Uses of XML

A report produced by the CEN/TC 251 task force for XML suggest the uses of XML in the healthcare area are as follows [Dude98]:

- Browsable reference materials
- A syntax for EDI messages
- XML content within EDI messages
- Publishing a record from a database for external browsing
- Publishing a record produced by merging different sources of data to allow browsing
- Representation of records in an archive
- Storage of records within individual systems
- Storage of common (distributed or central) record
- A format for inter-program communication

As can be seen from the list presented there are a number of differing ideas for the ways in which to use XML. However the viewpoints can be split up into several broad areas; document oriented, message oriented and EHCR oriented.

3.6.1.1 Document Oriented

It would seem that one of the strengths of XML is that of presentation of information in the same way that HTML is utilised on the web. Considering this further it would seem that the presentation of information in different ways to people with different access rights to a medical system based on an underlying architecture is a way forward. In this way sensitive information could be filtered so that only users see the information about the patient that is relevant to their access rights.

3.6.1.2 Message Oriented

The main likely area for the use of XML is using it in the area of EDI. It would be useful to utilise XML as an exchange syntax in the same way as Abstract Syntax Notation 1 (ASN.1) can be utilised. However the messages that are defined would have to be based on an underlying architecture.

3.6.1.3 EHCR Oriented

The third perceived area of use for XML is as an EHCR. This would be difficult as there is no way in which to utilise an underlying architecture on which to base the information held by the EPR. If there is no architecture issues such as the context in which individual items of data cannot be stored the administration information such as date and time the data was recorded, who entered the information and who is taking responsibility for it cannot be added to the information.

Also any term set information cannot be recorded without the full term reference being held with it.

The transfer of data cannot take place, as an extract of data cannot be defined. So again contextual information is lost.

No indication of how the patient is identified is given which is obviously a very pertinent area when it comes to medical records.

The *Techniques and Methodologies Working Group* (TMWG) have the following view of XML:

"TMWG believes that XML technology can be one of many types of functional service view implementations. However, the use of XML within an OO-edi environment would require a data transformation to map to business objects. Pure OO-edi using

distributed object technology does not require data transformation or mapping and thus is a more efficient solution. TMWG is continuing to conduct research on XML to determine its role, if any, in the EDI environment." [TMWG]

3.6.2 XML Conclusion

The XML/EDI group [Hinc98] was formed in 1997 and produced a document in January 1998 entitled *"Guidelines for using XML for Electronic Data Interchange"* [Brya98]. Areas that are being addressed by this group include the integration of web-based messaging with conventional EDI, global tag repository as well as sophisticated message validation.

It can be seen that the work in this area is in its infancy and any developments that come from it promise to be interesting. The reason that the work presented here did not follow this route is due to the infancy of the XML area.

In conclusion XML was developed for the presentation of data and the exchange of such data, not for the structuring of the medical record itself and these are the areas that it is best suited to.

3.7 CEN TC/251 WG 1

3.7.1 CEN TC/251 WG1 Overview

The *European Committee for Standardisation* (CEN) *Technical Committee* (TC) 251 *Working Group* (WG) 1 has recently set up four new *Project Teams* (PT) to look at certain aspects of the EHCR, these comprise:

- PT – 26 Extended Architecture and Domain Model [PT26]
- PT – 27 Domain Term List [PT27]
- PT – 28 Distribution Rules [PT28]
- PT – 29 Messages for the Exchange of Record Information [PT29]

3.7.2 PT – 29 Scope

The specific area pertinent to the work presented in this thesis is PT – 29 Messages for the exchange of Record Information. The scope of the work undertaken is to allow information to be exchanged between healthcare parties responsible for the provision of clinical care to an individual patient. The messages that have been defined allow

data held by one healthcare professional to be transferred to another healthcare professional.

In particular the messages can be used to convey:

- A complete copy of the patient's notes, stored on one system,
- Part of the patient's notes that form a logical extract,
- Parts of the patient's notes for updating of a parallel system.

The project team felt that two distinct properties of electronic health record communication were important. Firstly, that the communication of the information should be rendered human readable by the receiving system and secondly that the information that is being received should be processable by the receiving system.

3.7.3 Recommendations of PT – 29

This preStandard highlights the different type of communication scenarios that could take place between healthcare professionals. The scenarios do not form an exhaustive list but serve as examples of the type of situations that may arise, the scenarios presented are:

- Transfer initiated by an EHCR Source
- Transfer of care initiated by an EHCR destination
- Provision of a temporary service without a request from the EHCR Source
- Provision of a temporary service following a request from the EHCR Source
- Provision of continuing care by two or more partied
- Scenarios involving a third party

Three messages have been defined, which are:

- Provide EHCR Message – This is used to communicate all or part of a single patient's EHCR in response to a request EHCR message or some other means
- Request EHCR Message – This is used to request all or part of a single patient's EHCR
- EHCR Notification Message – This is used to enable the communicating parties to inform one another about the state or progress of EHCR communication.

3.7.4 Criticism of PT – 29

It is unfortunate that the project team found that it was necessary to define an architecture themselves, this was the remit of PT – 26 the results of which have been found lacking.

The document does not include any rules or guidelines on which to base the actual creation of a message or sending of a message. A company wishing to implement the messages defined has two options open to them. Firstly, define and develop proprietary messages, agree these messages with all parties with which they choose to communicate (this seems to defeat the whole purpose of having a standard in the first place). Or secondly, be forced to use the XML expression that is defined in the annex of the document (whether or not they agreed with its entirety or not). This is down to the fact that PT – 29 have been forced to focus most of their resources at the development of a model of the architecture.

There are no guidelines for the handling of data from legacy systems. Companies are not going to be in a position to change their systems overnight so that they are in accordance with the pre-Standard. Clear and precise guidelines should be presented to show how to handle this legacy data.

3.7.5 Conclusions

Whilst three messages have been defined the guidelines needed to implement them in a real system (other than the XML expression) have not been provided.

It is interesting to note that EDIFACT is not mentioned in the document, not as a possible syntax or as a syntax that has been used in previously defined standard messages. This omission serves to highlight that the members of PT – 29 have reached the same conclusions as have been reported in Chapter 2, albeit at a much later date.

The work that has been carried out by PT – 29 appears to be in line with the conclusions that are presented in the remainder of this work, in chapters 4, 5 and 6. However the work in chapters 4, 5 and 6 also deals with the problem of data in legacy systems.

3.8 CORBA™

3.8.1 The Object Management Group

The Object Management Group (OMG™) was founded in 1989 as a non-profit making organisation. Its goal was to develop technically excellent, commercially viable and vendor independent specifications for the software industry. The consortium has now grown and includes over 800 members. The main purpose behind the OMG is to define industry guidelines and detailed object management specification to provide a common framework for application development. By undertaking this task conformance to the specifications defined will allow heterogeneous systems to communicate information. The specifications already developed by OMG allow information interchange interfaces for distributed object computing. These standards are used around the world to develop and deploy distributed applications for manufacturing, finance, telecommunications, electronic commerce and healthcare.

Applications can communicate with each other by adhering to the specification of the Common Object Request Broker Architecture (CORBA™). CORBA 1.1 was introduced in 1991 and defined the Interface Definition Language (IDL) allowing client/server object interaction within a specific implementation of an Object Request Broker (ORB). In 1994 CORBA 2.0 was adopted defining true interoperability by specifying how ORBs from different vendors can inter-operate.

In order to transfer objects between different systems an ORB is used as the middleware. A client can transparently invoke a method on a server object if they both conform to the CORBA specification. The client and server can be on the same machine or distributed across a network. The ORB provides interoperability of objects independently of the programming language or operating system. The ORB provides interoperability between applications on different machines in heterogeneous distributed environments and seamlessly interconnects multiple object systems.

3.8.2 CORBAmed

CORBAmed is the healthcare domain task force that has been set up to specify object-oriented interfaces between healthcare related services and functions, in order to provide compatibility to a wide range of software components.

Current CORBAmed activities include:

- Roadmap
- Personal Identification Services
- Clinical observation access Service (COAS)
- Decision support services
- Lexicon query service
- Security
- Pharmacy

3.8.3 Review

The way forward in the exchange of information would seem to be by developing applications that are CORBA compliant. However, whilst working in this area developing an effective method for the exchange of information and at the time of writing there is not a complete specification with which to be compliant.

3.9 Summary

This chapter has undertaken a review of other methods for the exchange of healthcare information. The areas looked at were Health Level 7 (HL7) version 2, Accredited Standards Committee (ASC) X12, Synapses, Synex, Extensible Markup Language (XML), the work of CEN TC/251 WG1 and CORBA™.

Whilst some of the formats were deemed to be unsuitable for the exchange of comprehensive healthcare data, others were thought to be of interest in the future. One such area is the work being undertaken in the usage of XML. It was thought as long as the work continued within the remit of transfer and presentation, and not the storage of data it could be useful in the future. The other area to follow closely in the future is the work being carried out by the CORBAmed group. However at the time that research into this thesis was being carried out there was no effective, usable way in which to transfer comprehensive medical data in a generic way.

Chapter 4

4 GEHR Exchange Format

4.1 Chapter overview

It has been shown in the previous chapters that the methods adopted for the exchange of healthcare data, whilst a good first step, are by no means comprehensive enough for the wide range of data types and complexities of medical data [Domb96] that exist in the healthcare arena. This chapter gives an overview of the *GEHR Object Model* (GOM) and defines an alternative method to transfer all types of medical data

4.2 Introduction

The proposed solution - a transfer mechanism for the exchange of *any* medical data - is based on an emerging European standard architecture specifically designed for the handling of medical information. The *Good European Health Record* (GEHR) [Gehr95] is a project to come out of the *Advanced Informatics in Medicine* (AIM) initiative. The results and ideas from this project are feeding directly into the work of standards bodies and other National and European medical record projects.

One of the main deliverables of the GEHR project was an Object Oriented (OO) model for the *Electronic Health Care Record* (EHCR). The resulting architecture satisfies requirements that were defined during the early stages of the project. These included the requirement that any medical record should be comprehensive, communicating and portable [GEHR93c].

The resulting mechanism for the transfer of EHCR data that has been devised as part of this work is known as the *GEHR Exchange Format* (GEF), and is expressed in *Abstract Syntax Notation 1*(ASN.1) [Neuf92].

As the GEF is based on the GEHR Architecture it satisfies the basic GEHR requirements that means that the resulting transfer mechanism is indeed independent of a tr ansfer syntax, thus satisfying the CR1300 report [CR1300].

4.3 The Good European Health Record Project

In this section the GEHR philosophy is introduced as well as the concepts of the GOM.

4.3.1 The Electronic Healthcare Record

4.3.1.1 Definition of the Electronic Healthcare Record

The EHCR is the electronic record for one patient on one system (which will be termed an EHCR SOURCE). There is only one EHCR for each patient at this EHCR SOURCE. Everything that is contained in this EHCR is deemed to be about the patient. This aspect of "being about something or someone", which embodies the idea of 'data subject', is called the 'Scope' of the data here.

4.3.1.2 The Boundary of the EHCR

A view adopted by some people is that of the global healthcare record. They propose that the EHCR should comprise all of the information held on an information system pertaining to a patient, including components such as decision support and the process model [Cair91] of the institution. Others, notably clinicians, feel that the clinical record must be clearly defined, and that information should not form part of the record until a clinician has taken responsibility for that information and placed it into the record. This latter view requires that information created or received by the information system must only be considered part of the EHCR when a responsible clinician has authenticated it. For example, a laboratory test result might initially be held on a laboratory information system. It should not be regarded as part of a patient's healthcare record unless there is an entry, authored by a clinician who has responsibility for that patient's care, which contains that data (or an electronic reference to it) and any appropriate consequences for that patient's clinical management. In many ways this approach resembles that currently adopted for paper records, and mirrors a process that protects both patients and clinicians. This Specification proposes that there should be a clear border to the electronic healthcare record. The process model view of the EHCR is outside the scope of the GEHR view of the EHCR.

The implementation of electronic healthcare records might follow one or other of two quite different strategies:

- to mirror the concept of the paper record
- to create a new concept of a virtual, distributed healthcare record.

The second approach arises from progress in the fields of database management systems and networking. With the developments in telematics, it could be envisaged there would be only one, single, distributed, virtual healthcare record for each patient, representing the aggregate of all healthcare data of individual patients. Different healthcare professionals (HCPs) could then have specific access and views of such data, according to predefined sets of rules for access rights and other safety and security measures.

In the former approach, on the contrary, the EHCR is a tool used by one HCP (or by a team of HCPs) to manage individual patient data. In this concept, rather than one virtual record, several records may well exist for each patient. A patient could, for example, have one record kept by his/her local General Practitioner (GP), and one kept at the local hospital. Thus, as with the paper record, data is selected, organised, and authorised by a HCP to be entered in one Health Care Record (HCR) while responsible for the care of one patient. This concept of a personal and personally managed record (or one shared at the level of the local team) is implicit in many expressions of the extensive requirements researched and documented by the GEHR project. For example:

- The rationale for clinical decisions must be apparent from the record (what was done and why).
- The clinician in charge must check the results of investigations before they are committed into the record.
- The record should be structured in a way that preserves the original meaning of the information.
- The record must not impose the values of one society on the clinical practice of another.
- EHCRs must accommodate both highly structured methods of recording information and very informal methods of recording information.

It is apparent that healthcare professionals require local, flexible, highly adapted electronic healthcare records. The GEHR architecture for EHCRs has just such characteristics.

4.3.1.3 The Role of the EHCR

The healthcare record is an important tool supporting quality in clinical care. Just as there will be many different situations in which it is accessed, the record can play many roles in the provision of care to individuals and to populations. The following

list for the roles to be fulfilled by the record and given in [GEHR95] are based on a list originally proposed by Shortliffe & Barnett. The EHCR Should:

- Form the basis of a historical account
- Record preventative measures
- Support communication
- Remind clinicians about anticipated health problems and planned actions
- Identify deviations from expected trends
- Provide a legal account
- Support clinical research
- Enhance efficiency of health professionals
- Support continuing professional assessment
- Support medical education
- Accommodate decision support
- Access medical knowledge bases
- Assist with audit
- Accommodate future developments

The growth of national health services throughout the world has placed new demands on the healthcare record beyond that of the initiating clinician-patient consultation to include use by many interested parties. These include:

- the patients themselves and their appointed carers
- the clinician, in preventive or anticipatory care roles
- groups of clinicians working in primary or secondary care
- paramedical colleagues working with the patient
- clinicians and clerical or research staff for clinical audit, personal or department
- quality assurance
- hospital managers and healthcare purchasers (health authorities or insurers) for quality assurance
- healthcare planners at hospital, practice, district region or national level
- legal advisors for the patient or clinician
- clinical researchers
- medical students and medical teachers
- commercial product developers for market research (e.g. pharmaceutical industry)

- insurance companies for determining payment, or assessing risk
- politicians and health economists (and journalists!)

It is also important to bear in mind that the truly useful record retains its usefulness and integrity for the lifetime of the patient. This means taking steps to ensure that the EHCR data can outlive the electronic system within which it is stored.

4.3.1.4 The Structure of the EHCR

In technical terms, the EHCR is the top-level containment structure, and would be composed of one or more Transactions, together with some data enabling the record to be identified, see figure 10

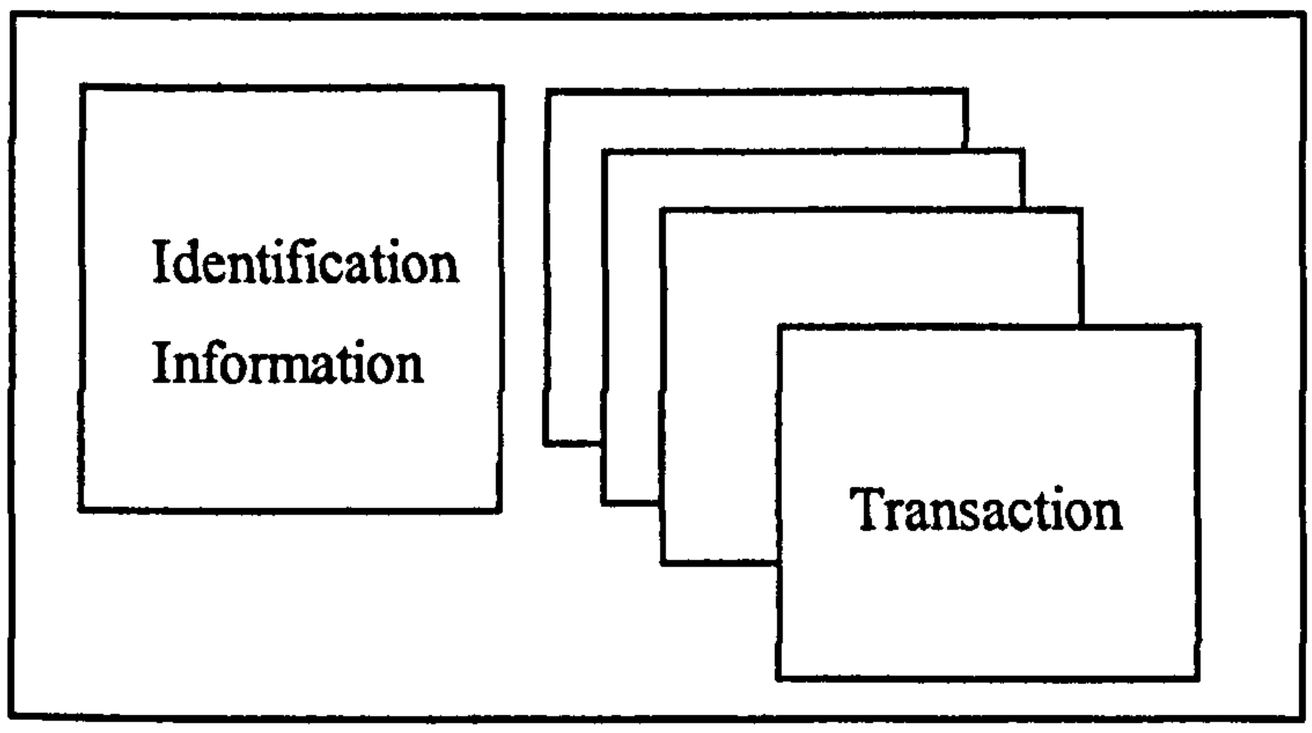


Figure 10

The EHCR itself represents the healthcare record for a patient, in electronic form, and is the central concept of the GEHR information model. The point in time when the EHCR began its life (a medico-legal requirement) is identified. It is possible for instances of EHCR for the same patient to exist simultaneously at various sites, due to care being provided by different facilities. The logical EHCR for a patient would be the result of merging all EHCR instances in the GEHR context, pertaining to the same patient. This is sometimes called the 'Virtual Record'. There may be any number of EHCRs for an individual, at different EHCR sources, but only one at each source, remembering that there may be a number of EHCR sources at a site.

A GEHR-compliant EHCR Source should not be confused with a Health Care Facility (HCF). Physically, an EHCR Source may correspond to a single computer, or to a whole network. As the server of EHCRs, the EHCR Source is the appropriate place to include semantics for the exchange of records.

All EHCR Sources are part of an owning HCF. The name of the EHCR source must be unique within the enclosing context.

The EHCR Extract abstraction is structurally the same as an EHCR and is intended as the form in which an EHCR is transferred to another site.

In the case where an EHCR (or EHCR Extract) is sent to a site at which an EHCR for the same patient may already exist there is a need to ensure that the records are reliably identified as being for the same patient. Although much work has been carried out in the area of the identification of a patient there is still no global patient identifier, the conflict may be resolved by comparing the latest version of the Patient information with that incoming. If there is any doubt, the final decision must be left to the person responsible for accepting the record at the receiving site.

4.3.1.5 The Transaction

A key clinical requirement is the ability to record details of each clinical encounter as a special grouping of items for medico-legal reasons. This grouping - the Transaction - is fully documented in [GEHR93c], where a Transaction is defined as:

"the information recorded about a patient by a single author in one institution at one point in time".

It represents the data entered in one interactive session with a patient record. This could result from a consultation or other contact with a patient, or perhaps from the 'filing' of a test result or letter.

The GEHR concept of "Transaction" should not be confused with the database management system notion of physical Transactions. A Transaction, in the GEHR context, corresponds to an interaction with the EHCR by one HCP at one point in time - that of committal. Although more than one HCP might be involved in creating the information in a Transaction, only one HCP commits the Transaction to the record. This is the authorising HCP. The same Transaction can never be committed again in the original or any other instance of an EHCR.

Seven different types of Transaction have been identified as follows:

- Contact – used to record information about an encounter with the patient
- Admin. – records the administration details of a patient
- Report – information recorded in the EHCR without the patient being present

- Summary – related to the past care that a patient has received
- Continuing Care – the description of future care planned for the patient
- Nota Bene – used to record information that is deemed important to be seen when accessing the notes
- Trigger – the place in which actions may be recorded as a result of various conditions being true

It is important to note that Transactions do not contain other Transactions.

Since this work was undertaken the notion of transactions and how they should be modelled has evolved (see section 4.4.1).

4.3.1.6 Unit of Transfer

In order that the EHCR may grow logically and in a way that preserves its integrity, the Transaction forms the basic medico-legal unit of the clinical record. The Transaction is the minimum grouping of data for the communication of healthcare record data. Note that any healthcare record data communicated must always contain, with the Transaction(s), sufficient identification information of the patient so that the information can be added to the patients record already stored or for the creation of a new record should the record not be found. Unambiguous identification is very difficult, but it should always be possible to give a level of certainty of identification. The major aim is to avoid erroneous identification. It is recognised that it is possible for instances of EHCR for the same patient to exist simultaneously at various sites. This may occur when the patient is being given care at two healthcare facilities e.g. at a hospital and by a General Practitioner. The *logical* EHCR for a patient would be the result of merging all EHCR instances that pertain to the same patient.

4.3.1.6.1 Dealing With Mistakes

Transactions are permanent. Once committed by the appropriate HCP, they may be amended – to correct mistakes – but not erased. A formal amendment concept based upon tried and tested versioning schemes has been established for Transactions where a "Versioned Transaction" contains all its versions that result from formal amendments. This is necessary to cater for correction of errors in the recording of healthcare data. For example, if a HCP has committed data to a record that is later found to be in error, the error must be amended but the fact that erroneous data was at

some stage held, must remain. Clinical decisions based upon the erroneous data may have been made and an accurate audit trail is vital in, for example the future care of the patient or in a case of later litigation. An amendment will result in an additional version within an existing Versioned Transaction, whereas the addition of new information always results in a completely new Versioned Transaction.

It is not envisaged that many different versions of a transaction will routinely exist.

4.3.1.7 Health Record Item

While data can be entered in EHCR in many different formats (reports, laboratory result sheets, forms, etc.), it has proved useful to define an elemental unit of data entry: this concept of the smallest unit of information which remains meaningful as an entry in a HCR is seen as fundamental. The name used here for this construct is the Health Record Item or HRI. Other names have been used for this type of construct within the HCR - the fundamental concept is widespread. Traditionally, individual patient records are built by adding entries at the appropriate location in the relevant record. The way these entries are grouped adds to their meaning. HCRs are collections of Entries (Observations, Headings, etc.) which are progressively accumulated as the history of the individual concerned evolves in time. In paper records data may be entered in free text or onto a specific form or report inserted in a given place in the folder, which represents one patient record.

In electronic records there is much wider scope. Electronic systems often use the concept of HRIs in one form or another, although very often the specific structures chosen are not very flexible or amenable to change which causes problems with advances in both information technology and medicine. Another cause of problems in many cases is the lack of a Transaction concept. Without this, portability and the maintenance of integrity over time becomes very difficult.

The HRI provides the mechanism for expressing the content value of Entries made in the record.

At the logical level a HRI can be regarded as the unit of information that can be obtained as the result of one specific measurement, question, observation, discussion, or other investigation mechanism. For example,

patient's weight = 80kg

is logically a HRI. However, healthcare data is not always as clear cut as this and it is vital (for comprehensive recording, maintenance of integrity and so on) that recording systems can cope with HRIs such as:

> *patient's weight = 10 st but was measured on an old mobile scale which may not be reliable. If accurate, this recording shows a worrying increase.*

The HRI was adopted by CEN TC/251 (PT011) as the basic unit of health information within the record and is referred to as a Record Item (RI). It represents the finest granularity by which an individual piece of information may remain meaningful if viewed in isolation (although complete interpretation may require it to be seen in perspective with other related Items - the clinical context). In essence, the HRI is composed of an Item Name, its primary content value, and other associated identifiers, properties and attributes.

In paper HCRs, instances of HRIs derive their meaning from their constituent elements and from the context in which they are recorded:

- they have two main constituents:

  - an identification (or name);
  - a content (or value);

  but also gain meaning from such as underlining, circling, a scribbled comment in a margin and so on;

- they represent characteristics of the data subject;
- they derive some of their meaning from the higher level structures to which they belong – the position on a paper pro-forma for example.

The main content of a HRI can be one of a wide range of data types, including dates, text strings, longer narrative comments, numeric values, and multimedia data types such as images and biosignals. Some HRIs may also have, as a content, a code referring to a given coding scheme (e.g. a diagnosis expressed as an ICD9 code, or a drug expressed as a Read code).

4.3.1.8 Health Record Item Collection

The HRI Collection provides a mechanism for narrowing the Scope of the data. HRI Collections may contain other HRI Collections and HRIs. The lowest level of HRI Collection contains only HRIs. HRI Collections with their subordinate HRIs and/or HRI Collections are used to express the component parts of clinical concepts in the correct structural relationship appropriate to the clinical concept, and to assign values to their component parts.

The term *HRI Collection* is used here to indicate a structure that contains groups of Observations. HRI Collections allow for the construction of complex aggregations of data. Examples might be:

```
HRI_COLLECTION=Urea and Electrolytes
  HRI=Potassium
   content=4.3 mmol/l
  HRI=Sodium
   content=140.0 mmol/l
  HRI=Calcium
   content=2.41 mmol/l
  HRI=Alkaline Phosphatase
   content=95 iu/l
  HRI=Creatinine
   content=75.0 µmol/l
  HRI=Phosphate
   content=1.170 mmol/l
  HRI=Urea
   content=3.1 mmol/l
  HRI=Albumin
   content=43.0 g/l
  HRI=Total Protein
   content=75.0 g/l
  HRI=Total Bilirubin
   content=7.0 µmol/l
```

The recursive structure of the HRI Collection allows the HRIs to be assembled into completely flexible structures.

HRI Collections derive their meaning from their constituent elements and from their context.

- They have two main constituent elements:
  - an identification (or name);
  - Observations (HRIs or HRI Collections);

- they group observations on the patient of whose record they are a part;
- they derive some of their meaning from their *clinical context.*

The HRI Collection is similar to the CEN TC/251 (PT011) Health Record Item Complex in the CEN preStandard 12265. However, CEN has not yet distinguished between the two concepts of Collection and Heading (as described in [GEHR95 section 5.4]), and uses the HRI Complex for both. CEN have therefore found it necessary to specify an explicit data subject attribute. The scope rules of the HRI Collection lead to the unambiguous definition of the data subject of a group of observations, and no explicit data subject attribute is required.

4.3.1.9 Heading

The Heading provides a means of grouping or labeling combinations of Collections/HRIs. It allows instances of clinical concepts, expressed through Collections and HRIs, to be related to the context of healthcare (and its recording) for the patient. This property of labeling or grouping is called Annotation in [GEHR95], clearly to distinguish it from all other combinational devices. Headings do not narrow the Scope of the data.

An example of a heading is given below:

Heading = Biochemical Tests
HRIC = Urea and Electrolytes
HRI = ...

Figure shows the relationship between Headings (H), Collections (HC) and HRI's (HRI).

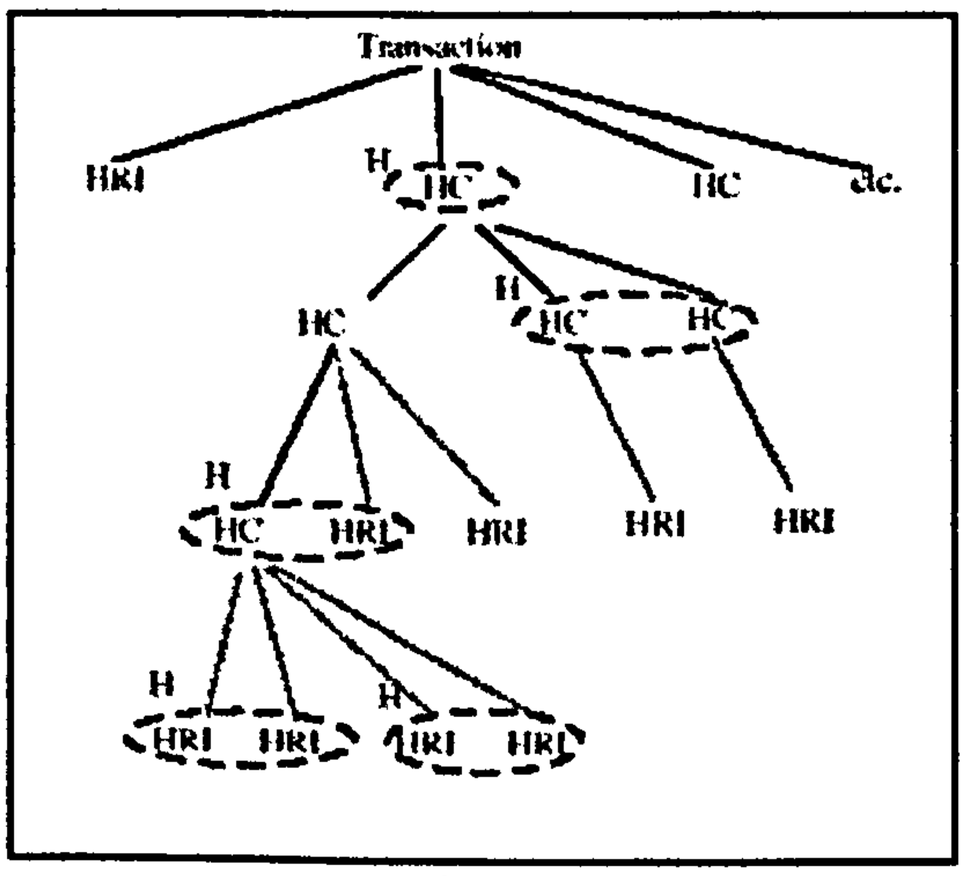


Figure 11

The splitting of a clinical recording into its component parts must be done, where appropriate, at the architecture level, to retain integrity and avoid ambiguity as the record progresses through time, from system to system over the lifetime of the patients EHCR.

[GEHR95, section 4.3.8] gives some guidance on how the choice between HRIs, HRI Collections, and Headings is made to represent any clinical concept and its relationship to the patient in an EHCR.

4.3.1.10 Attributes

Each of the above constructs has attributes defined in the Model for capturing the necessary identification, content, and context of the Entry. The term "context" is used for a category of characteristics of the Observations, which have several features in common:

- they are not essential in identifying an Entry;
- they can be shared by several Entries in the same record (e.g. several measurements can have the same date, the same person responsible for making the Observation);
- they usually refer to the context in which an Observation has been recorded.

Example characteristics include:

- context of the provision of healthcare:

  - person responsible for obtaining/providing the information
  - date/time observed;

- ethical/legal context of the data:

  - person responsible for recording the Entry;
  - access rights;

- clinical interpretation of the Entries

  - degree of certainty of Entries;
  - links between individual Entries - general / problem, etc.

- presentation of the Entries

  - organisation of the Entries;

- emphasis;
- language of recording.

Secondary operations may occasionally be performed on the data within a HCR. Such secondary operations may include linking data together (e.g. problem links), adding emphasis (e.g. things not to forget...), summarising, etc. Although no new data are added, creating new relations between the data provides new information. The data can be viewed according to the initial structure, or according to other structures emanating from these links.

Figure 12 below summarises the main points:

Principal GEHR architectural components

- the **EHCR**

  provides the container for all data about a particular patient

- the **Transaction**

  provides most of the features needed for the medico-legal aspects of healthcare data

  provides the mechanism for the control of amendments

  represents the smallest amount of data which can safely be transferred between EHCR systems

- the **Health Record Item (HRI)**

  provides the structure for recording the content values of EHCR entries

- the **HRI Collection**

  provides for aggregation of HRIs and other HRI Collections

  provides the means of changing the scope (data subject) of the data

- the **Heading**

  provides annotation for groups of HRIs/Collections

Figure 12

4.4 The Good European Health Record Object Model

The *GEHR Object Model* (GOM) (Figure 13) describes formally the classes and relationships between classes that have been designed for the medical record. It aims to contain all data fields that are needed by HCPs both at present and in the future, or mechanisms for dealing with data that it does not recognise.

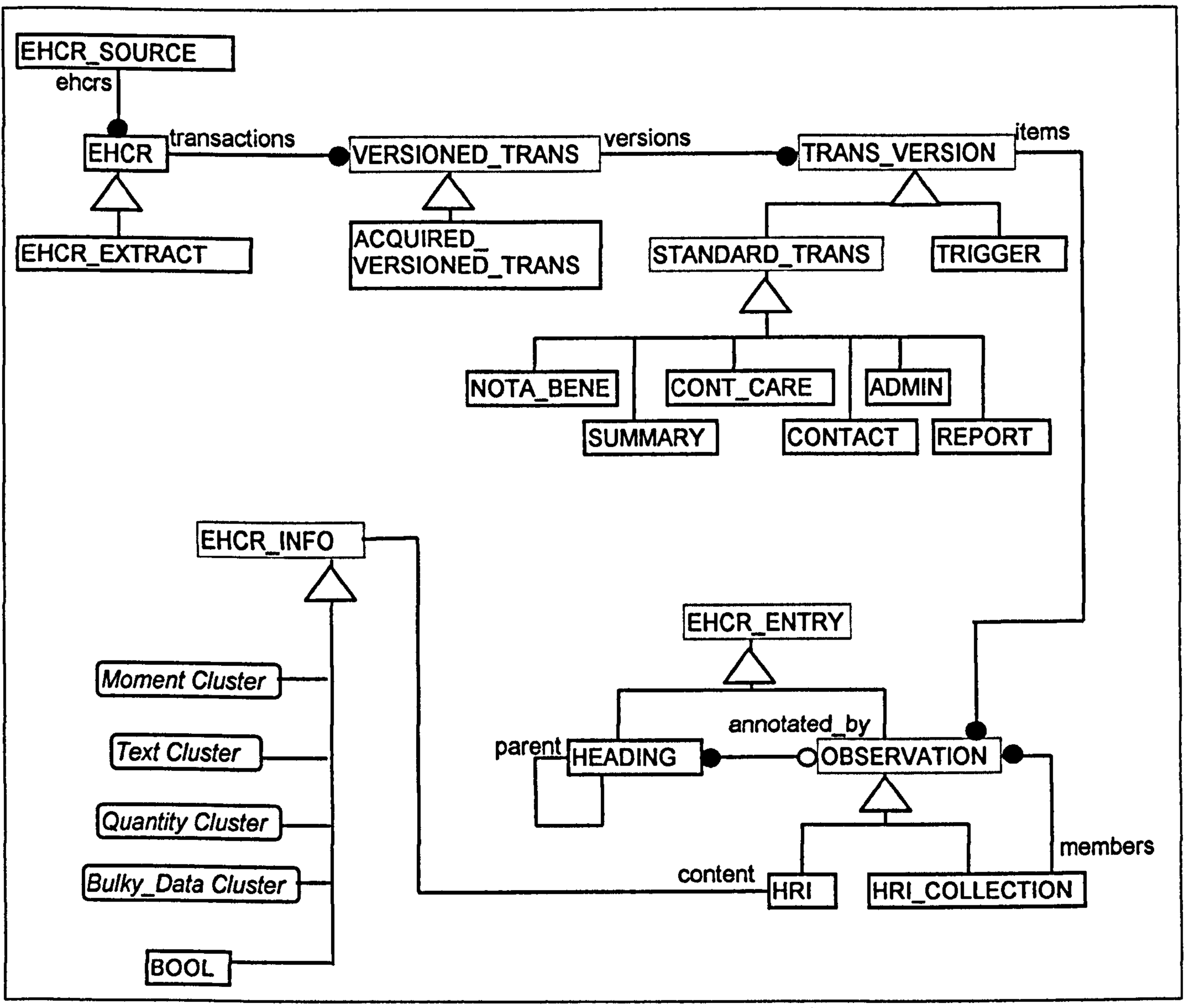


Figure 13: Abridged GEHR Object Model v1.0

The GEF provides an extract of data from the EHCR of a patient on one system for transfer to another. The logical ('global' or 'virtual') EHCR for a patient would be the result of merging all EHCR instances in the GEHR context which pertain to the same patient.

Patient data that has been recorded during the same encounter with the patient, would - in GEHR terms - be part of the same transaction: more specifically the Contact Transaction, together with information about the clinician responsible for recording the data, a single date and time the data was collected together with the source of the information. It would not be permissible to transfer any part of this transaction, any single collection or HRI in isolation, nor without the details of the transaction itself i.e. the person responsible, date-time of recording etc. Because the information is expressed in GEF, no context is lost and the data is verifiable as complete.

4.4.1 Progress Since GEHR v1.0

Since the end of the GEHR project, work has continued which has inevitably resulted in the proposed evolution of the GOM. The present version, proposed by the EHCRSupA project, is now version 1.5. Whilst the look of the model has changed considerably the basic principles and ideas that fed into it have not. The latest diagrammatic version of this proposed model can be seen in Appendix C. For an in depth review of the decisions, discussions and ideas that were addressed during this evolution see [Dixo97b].

Following on from GEHR work has continued and expanded in many different EU framework projects and International working groups such as CORBA™. Both EHCRSupA and CEN are bringing this work together.

As has been stated, whilst the model has changed the fundamental principles have not, this means that the conclusions that were reached during the early part of this work, presented in Chapter 2, are still valid.

The rest of this chapter details the syntax used for the GEF, ASN.1 and describes in depth the GEF.

4.5 ASN.1 Design

ASN.1 is a mechanism for communicating entities of data between different computer systems. It provides for communication between heterogeneous systems e.g. between different computer environments, between applications that have been implemented in different programming languages, different application systems and heterogeneous networks. ASN.1 copes with these different paradigms, as it is an external data representation language that supports heterogeneous interconnection.

ASN.1 can be seen as a type of programming language which has built-in data types, a set of rules for constructing user defined types and a mechanism to set constant values of these types [Neuf92]. The built-in data types included in ASN.1 are *Integer, Real, Boolean, Bit-String, Enumerated* and *Null*. A user-defined type can be built up using these types to define any data type that the user wishes.

There are many structured types that can be used to combine the simple types into more complex types. These are *Sequence, Set, Sequence of, Set of* and *Choice*. Sequence and Set are similar in that they group together a user-defined named type. The difference between the two is that when using sequence the order of the types

defined is important when transmitting to another party, whereas when using set the order is not important as the elements of the user-defined type are tagged or numbered making each grouped type easily identifiable. An example is given below:

```
AdminData     ::= Sequence{
        Name            Bit-String
        Date-of-birth   Date
        Age             Integer }
```

```
AdminData     ::= Set {
        Name            [0]     Bit-String
        Date-of-Birth   [1]     Date
        Age             [2]     Integer
```

Sequence of defines a group of ordered types that are of the same type whereas Set of defines an unordered group of types that are the same.

The Choice construct gives the facility to define a type from a set of candidate E.g.:

```
Marrital-Status         ::= Choice{
                Single          [0]     Bit-String
                Married         [1]     Bit-String
                Divorced        [2]     Bit-String}
```

4.5.1 Encoding Rules

During the actual transmission the data it is in a format known as the *Basic Encoding Rules* (BER). ASN.1 allows for several different encoding syntax.

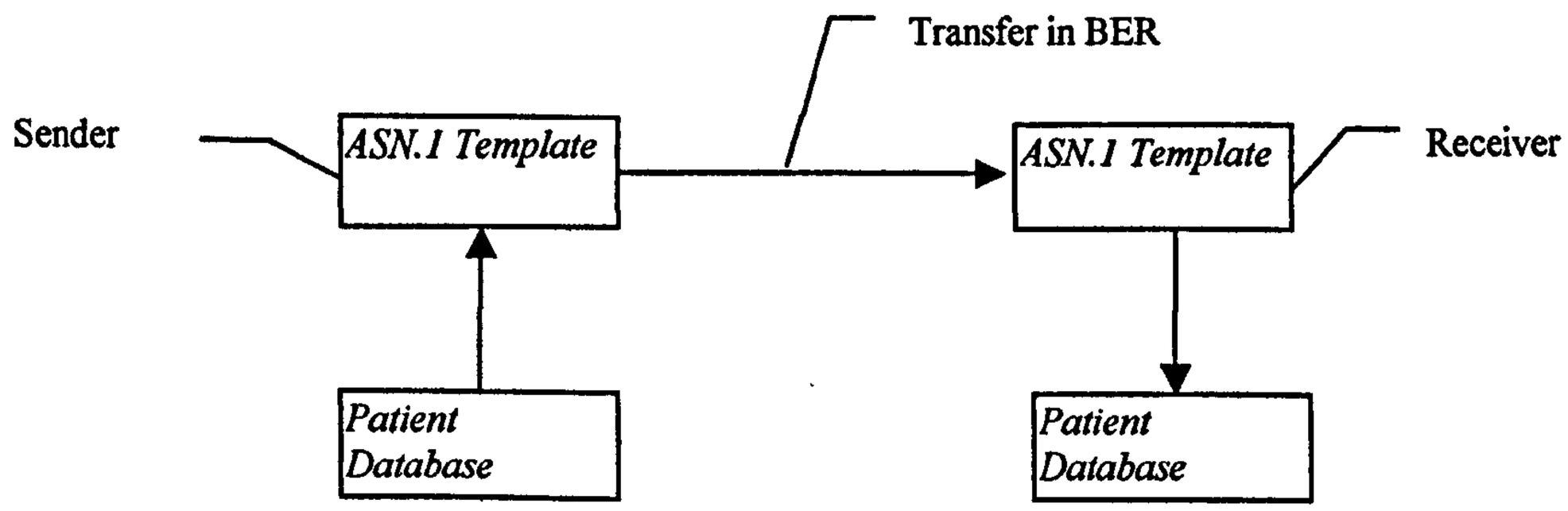


Figure 14

There are three main parts that represent the data type when it is in BER format: *Identifier*, *Length* and *Value*. The tags defined in ASN.1 for identification purposes are transferred with the value and because the type does not determine the size of the value, the length of the value is sent as well. If the data type is constructed then the value part itself will contain other types and values. See figure 15.

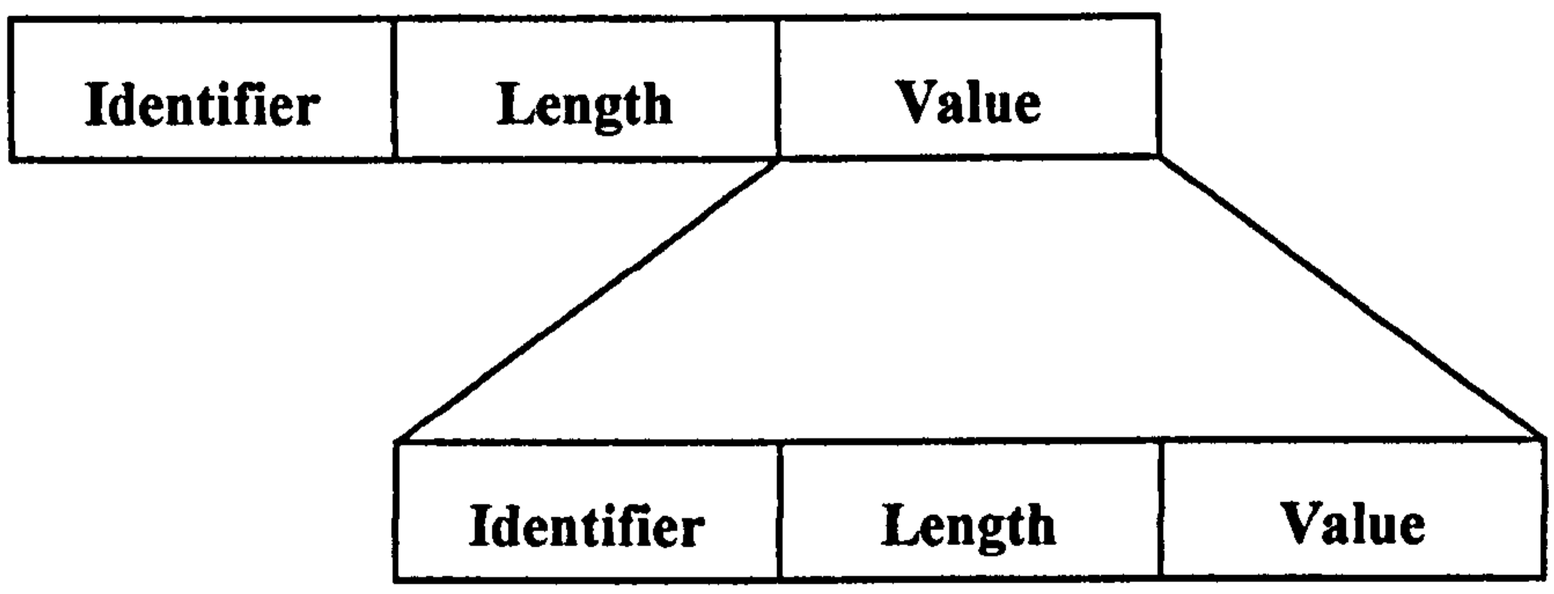


Figure 15

The identifier is made up of three parts: Class, Primitive/Constructed and tag identifier. The class can be one of four:

- Universal – Universally Defined
- Application – Defined for the type of Application
- Context specific – Defined within the particular context
- Primitive – One of the primitive types

Bits 7 and 8 in the identifier octet define the class. The primitive/constructor type within the identifier octet, bit 6, indicates whether the identifier is simple or constructed. Bits 5 to 1 are the tag number identifying the type.

The length of the contents is passed in the next set of octets. This can be described in short form, or one octet, when the length of the value is less than 128 bits in size, or long form when the length of the value octets is greater than128 bits in size.

The encoding is expressed in binary octets. The ASN.1 type definition AGE ::= Integer, with a value 25, would be encoded in primitive form as the built-in type UNIVERSAL 2, Length 1, value 25.

When all the types and values have been encoded the data format is in many octets which when put together form a bytestream that can then be transmitted. The receiver translates the message using the GEF as a template to recognise what they are receiving, using a translator they can then interpret the bytestream for use in their own medical database which may not be of the same type as that used by the sending system.

One area to be aware of is the large overhead of administrative data that has to be sent with the actual data values in ASN.1. The amount of administrative data sent when the actual amount of medical information being sent is relatively small may be quite large. This may be a deficit when communication lines are slow but is becoming less of an issue as technology advances and fast speed WAN's are set up, such as the NHSNet [Tele97]

4.6 GEHR Exchange Format

Using the GOM as a basis, the GEF can then be used to facilitate the transfer of data. The main difference between the OO model and ASN.1 is those internal identifiers and corresponding pointers have been used where the GOM uses one to many relationships.

The ASN.1 was derived from the GOM by hand, Appendix D. However it is envisaged that in future this will be done automatically to guarantee correctness, also for speed and efficiency when a new version of the GOM is defined. For validity purposes the ASN.1 was compiled, using the Snacc 1.1 compiler [Samp93], to produce C and C++ encoding and decoding routines. This compiler was used, as an ASN.1 to a more appropriate language could not be found at the time. A compiler producing routines in these other languages would have been desirable as the software tools generated from this work were written in Visual Basic.

The GEF is currently expressed in *Abstract Syntax Notation 1* (ASN.1), but can also be mapped onto any other suitable syntax, adhering to the guidelines set out in [CR1300]. The GEF can then be transferred as a byte stream using any medium available to both the sender and recipient. Additionally, the means of physical transfer is not specified. Any scheme agreed between the sender and recipient will suffice, providing that the data can be sent and received safely and securely.

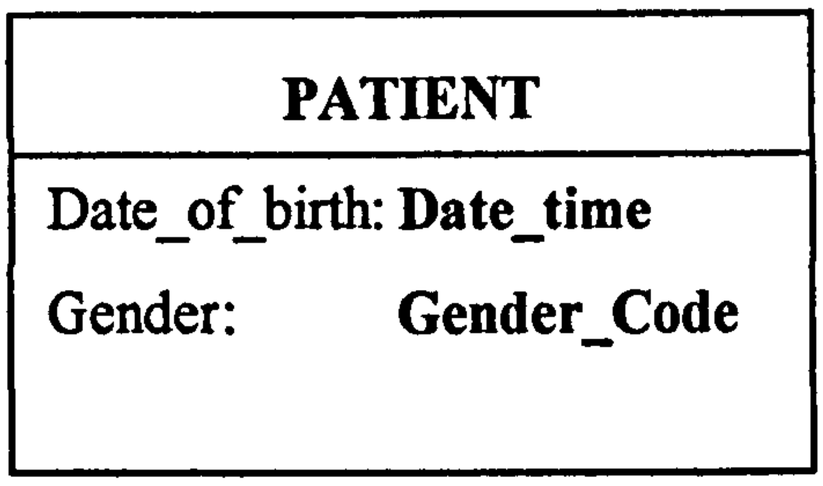


Figure 16: Patient as described in the GOM

The GEF was founded by taking each of the Classes and Attributes expressed in the GOM and describing them in ASN.1. An example of this is shown in Figure 17.

```
PATIENT       ::= [PRIVATE 72]
                  {                PERSON,
                   date-of-birth   [0]   DATE-TIME,
                   gender-code     [1]   GENDER-CODE}

DATE-TIME     ::= [PRIVATE 56]
                  {                MOMENT
                   dt-date         [0]   DATE
                   dt-time         [1]   TIME }

DATE          ::=  OCTET STRING (SIZE(8))
TIME          ::=  OCTET STRING (SIZE(6))

GENDER-CODE ::=  ENUMERATED
                  {male(0),female(1),unknown(2)}
```

Figure 17: Patient as described in the GEF

Each class is broken down into its constituent parts and these in turn are broken down to the lowest level. A full expression of the GEF can be found in Appendix D.

In overview the GEF achieves many things including attribution of the original information to the person legally responsible for entering it, as well as the date and time the data was entered. Any term that is not familiar to the recipient may also be transferred. Even if data is received of a type the recipient is not expecting they can view it.

4.7 The Exchange

The method so far described to facilitate the transfer of data is to use an EDIFACT message as a header followed by an ASN.1 based bytestream in BER format. As

detailed already this conforms to the NHS guidelines for data transfer [CR1300] i.e. using an EDIFACT message. Also, as well as being able to transfer data between GEHR compliant systems this method promotes the exchange of information from non-GEHR, *Legacy* systems, to GEHR systems. To do this the data in the legacy system is put into *Legacy Intermediate Format* (LIF) [Grub96]. The LIF is a simple text format that follows GEHR structures and guidelines, it specifies the obligatory data for each patient record. Once it is in this format it can then be encapsulated in GEF and transmitted to any GEHR compliant system.

4.8 Benefits of GEF

Using the GEF overcomes the problems that will be experienced when using the DIM and GMD [see Chapter 2]. Also the problems inherent in EDIFACT [see Chapter 2], as the GEF can cater for modification of the message while remaining compatible with the given standard, without needing a lengthy process to modify and disseminate a new message.

A mapping of data from existing ('*legacy*') systems, where the full richness of data is not present, to GEF can be defined, allowing information exchange that maintains the integrity and context of the original data. This allows for the transfer of data from any existing medical record system to any GEHR based system or any other legacy system, although how the receiving legacy system copes with much of the data it receives cannot be guaranteed.

The transfer of data from a legacy system to a GEHR compliant system has been undertaken and shown to work [see Chapter 6]. This demonstrates that the GEF copes with the data that is needed for transfer, adding all the relevant ethico-legal and contextual information that is needed for each data item, so that a GEHR based system can use the information that it receives.

4.9 Conclusion

In this chapter some of the concepts of the GOM have been presented to show how the GEF has been developed. The GEF, that is an exchange format for the GEHR architecture, is expressed in ASN.1.

In summary, introducing a new field e.g. an endoscopy video, would be technically challenging and take several years by the EDIFACT vehicle. Using the GEF, it could

be done immediately with no change to the message structure and without compromising the integrity of the data.

Systems built around the GOM will be able to maintain all the required information needed such as attribution, date and time data were committed to the record, and details of the clinical context. (In addition, the security and access controls afforded by the systems will sustain the integrity of this data). The record must contain (or reference) all information thought to be clinically relevant to the care of the patient and this clinical context must be faithfully maintained when communicating to a second party, and on for the lifetime of the patient record. The lifetime of the record may be longer than the lifetime of the patient, when considering the use of notes for statistical analysis and epidemiological studies.

Information transferred between systems based on a comprehensive healthcare architecture such as the GOM will not be subject to the dangers already mentioned provided that various principles are adhered to.

The state of the art in terms of the way in which technology has shaped ideas may have moved on since the GEF was conceived however the main principles behind the GEF have been proven. If innovations were to move away from this philosophy then problems such as those highlighted in Chapter 2 would be met. However if technology moves forward based on the underlying principles set out in this chapter the work undertaken shows that the exchange of information is following the correct principles.

Chapter 5

5 The Integration of Data in Existing Systems with GEHR Based Systems

5.1 Chapter Overview

This chapter addresses the issues pertinent to the area of integrating existing data with GEHR based systems. There exist a large number of diverse systems, both electronic and paper based, each of which contains a wealth of patient data. This data needs to be captured, so that it can be upgraded and put into new systems, making sure that the existing data is not lost. This data will then be available on systems that are far more flexible in the way they handle the data in comparison with *'legacy'* systems. There is a belief that data can be in some way captured in an ad-hoc fashion from legacy sources without loss of integrity. This chapter disputes this paradigm and presents a comprehensive alternative.

It is essential that future health information systems are capable of storing and communicating a wide variety of clinical and related information adhering to existing and emerging standards. The GEHR project developed an information model for electronic health records in Europe covering requirements for clinical comprehensiveness, portability, communicability and ethico-legal issues, see chapter 1. As has been shown in chapter 4, a method has also been defined for the transfer of such data, known as the *GEHR Exchange Format* (GEF).

There are a number of ways in which clinicians with data in existing, legacy, sources can migrate to GEHR based systems. One way in which existing system vendors may be encouraged to migrate is by the use of an intermediate but comprehensible method, in the way the data is structured. This chapter details the requirements for a *Legacy Intermediate Format* (LIF) as a means to transfer data from diverse legacy sources, be these paper or electronic. An example of this is also presented.

5.2 Introduction

There exist many healthcare systems throughout the world today that are completely different from each other in the way they store, handle and present clinical information [GEHR92, GEHR95]. The challenge that has been faced by computer suppliers over the last few years, and indeed will become a bigger issue in the future, is the communication and subsequently the integration of data held on these systems.

Projects are being funded, such as Synapses [Kalr96], to address the problems that might arise. However, what is needed for systems of the future is a common architecture upon which to achieve some kind of consistency [Dixo97a, GEHR95]. Without a common underlying architecture, data integration without loss of integrity is exceedingly difficult. With such an architecture, existing legacy systems can work towards compliance with the standard.

One such emerging European standard is the GEHR architecture [GEHR95] that provides an implementation independent information model on which to base the data held in healthcare systems. Following on from GEHR, the support action - EHCR-SupA[1], provided major input into the standard for Electronic Healthcare Record Architectures (EHCRA) being produced by Comité Européen de Normalisation (CEN).

5.3 Existing Patient Records

Records of patient information that exist at present are many and varied, ranging from paper to electronic recording systems. Although there is a potential wealth of information, accessing this data can be difficult. As Holland observed about the UK *National Health Service*:

> *"Despite huge expenditure on Information Systems in the NHS the information available to researchers remains poor. In part, at least, this is because the basic data are themselves poor" [Holl94]*

To make sure of the quality of data held in the future, systems being built should be based on a suitable standard such as that emerging from CEN. The requirement for such a standard is widely acknowledged [Rect91], [Mila96], the question is how soon it might be achieved. To sustain accessibility to data, it is essential that there should be a clear distinction between data and the systems within which they are held. Systems can and do change, but it is essential that the data they contain remain accessible.

[1] EHCR-SupA is a project under the European Union Framework IV Telematics programme.

Until now, when the decision has been made to change from one system to another for the recording of patient data, much data has in practice been lost [Hawk95]. It is vital to avoid this situation when moving to new standards based systems.

Legacy systems, especially paper-based, contain many years of historical patient information [GEHR92] which is very valuable and should be maintained. When upgrading to new systems a method of transferring data is needed. This method needs to be secure and it also needs to maintain the context. Contextual information can cover areas from ethico-legal information to date and time of recording or who recorded it.

5.4 Migration to New Systems

Not all system suppliers will choose to upgrade their systems to be based on the emerging standards. There will not be a plethora of new systems but people will still be required to integrate their data from legacy systems to the new systems as they emerge. In the past, changing from one information system to another has been a difficult (if not impossible) task and the extraction of data from one system to incorporate it into another has been subject to many problems. These difficulties will not apply to patient data held on future systems based on the standard since the data remain distinct and separable from the system.

Disparate systems wishing to migrate towards the standard may opt for the SYNAPSES route[2]. This project sets out to solve problems of sharing data between disparate information systems by providing the means to combine healthcare records consistently, comprehensively and securely through the development of a mediating server [Kalr96], [Grim96], [Tous96]. Whilst the SYNAPSES route maybe a good way forward, and appropriate for more sophisticated systems in the future, it is not yet finalised. A simpler method allowing not only data migration, but also a route forward for system developers, is outlined in the rest of this chapter.

[2] SYNAPSES is partly funded under the EU Health Telematics Framework IV Programme

As has been shown in chapter 4, the GEF is a transfer mechanism explicitly written

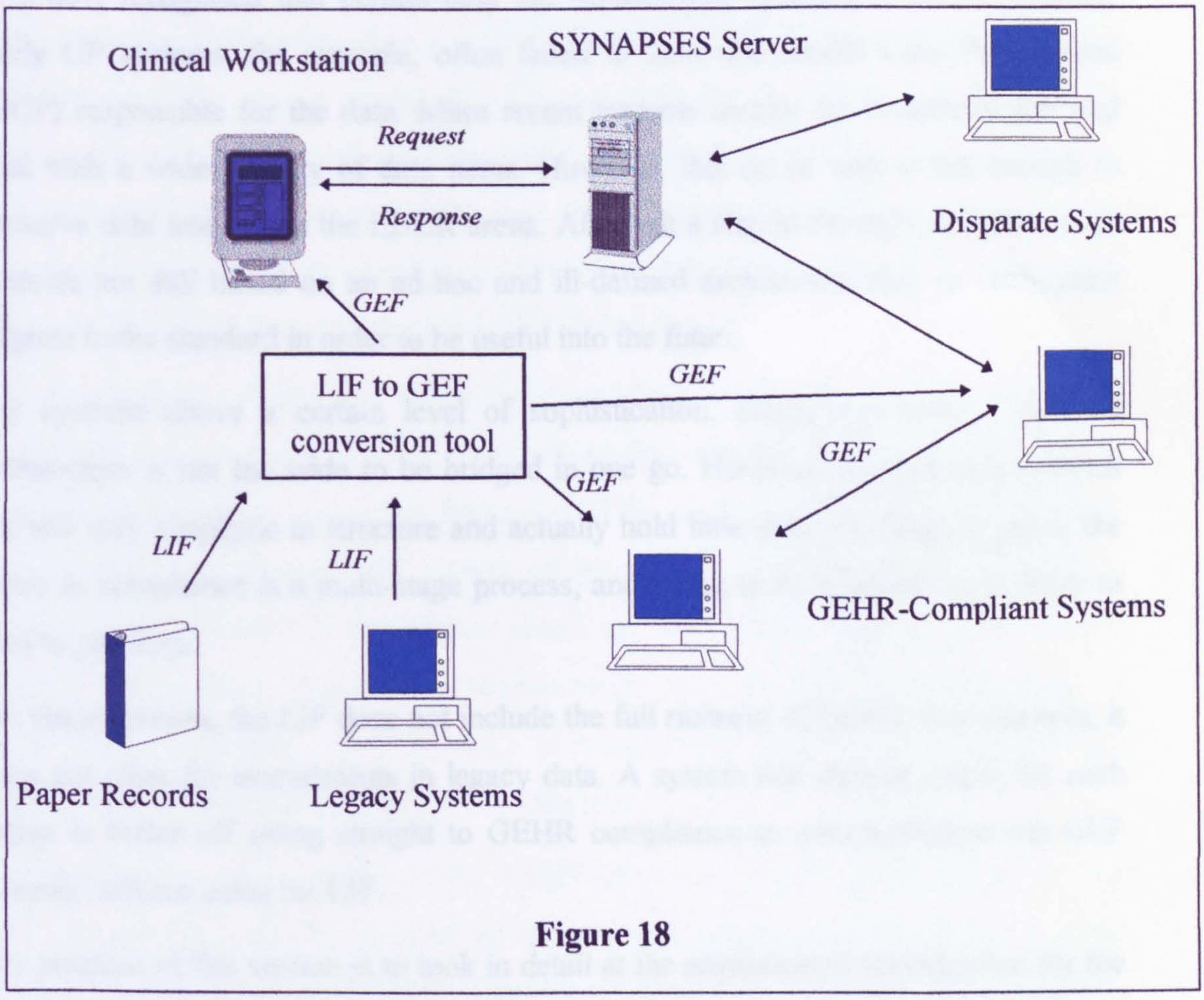


Figure 18

for the transfer of GEHR data. This is independent of the GOM and may be expressed in any suitably flexible notation [Elli96b]. The current version is expressed in Abstract Syntax Notation One (ASN.1) [ASN1].

The system suppliers not familiar with ASN.1 or the details of the GOM may benefit from the use of a more friendly intermediate expression. To this end, a Legacy Intermediate Format (LIF) has been developed. This is expressly for the purpose of capturing data from existing electronic and paper records to be used in - or to communicate with - GEHR compliant systems. LIF is intended as a half-way stage - a useful first step in organising legacy data which may currently be stored with little structure, context, etc., as a means of 'massaging' it into shape as far as possible without wrestling with the full richness or full technicalities of the GEF.

5.5 Legacy Intermediate Format

5.5.1 Introduction

It is now recognised that certain data are fundamental in effective record keeping. Early GP systems, for example, often failed to store the Health Care Professional (HCP) responsible for the data. More recent systems usually try to address this and deal with a wider variety of data items. However, this on its own is not enough to preserve data integrity in the EHCR arena. Although a step in the right direction, such systems are still based on an ad-hoc and ill-defined architecture and, as such, must migrate to the standard in order to be useful into the future.

For systems above a certain level of sophistication, compliance with a standard architecture is not too wide to be bridged in one go. However, many legacy systems are still very simplistic in structure and actually hold little data. For these systems, the move to compliance is a multi-stage process, and trying to do it in one go is likely to lead to problems.

For these reasons, the LIF does not include the full richness of GEHR. For example, it does not cater for amendments in legacy data. A system that already caters for such things is better off going straight to GEHR compliance or communication via GEF directly, without using the LIF.

The purpose of this section is to look in detail at the requirements specification for the LIF for data in order that it may be transferred between systems, particularly with regard to GEHR systems interfacing with non-GEHR systems.

The principles of GEHR that underlie the LIF structures used are fully explained in [GEHR95, section 5]. An overview is given of the whole data transfer process and the structures in the EHCR are described. The aim is to give the information required in the appropriate detail to allow LIF to be created from any non-GEHR medical database.

5.5.2 Background

Data that conform to the GEHR architecture will be comprehensive, portable, communicable, secure and flexible. It will have the ability to adhere to such ethico-legal constraints and requirements as may be necessary. It will allow the use of data

in any form, coded, uncoded, hypertext, multimedia, etc. as well as proprietary forms such as clinical drawings from specialist software packages.

Medical information systems that conform to GEHR will tend to hold more comprehensive information than is usually held in existing medical databases. For example, both the person responsible for any entry into a patient's record, and the person who actually entered the data will always be kept. So, if an entry is made in error and then corrected at some later date, the necessary information is kept such that the mistake can always be traced. The architecture allows for upgrades to term sets or full systems without loss or corruption of historical data.

The full range of GEHR, why it was developed (and is still developing) in the way it is and how it achieves its aims for the electronic medical record are topics beyond the scope of this chapter. For a full discussion of the background to GEHR, the GEHR requirements and the technical aspects of the GEHR architecture see "The GEHR Architecture" [GEHR95]. For further detail on the requirements, see the specific GEHR deliverables describing requirements for clinical comprehensiveness [GEHR92], portability [GEHR93a], communication [GEHR93b], ethico-legal [GEHR94a] and educational [GEHR94b] aspects.

People seeing a representation of the GEHR structured record [Appendix B] for the first time may be somewhat daunted by the apparent amount of detail stored. It is not easy to see the reasons and justifications for all aspects of the structures at a glance. However, it should be borne in mind that the GEHR architecture is the result of many man years of effort and investigation involving many different groups (clinicians, software engineers, quality assurors and so on) across Europe. The issues have been thoroughly thought through and tested. If, for example: the "obvious" path seems not to have been followed, there is a good reason, the "obvious" paths have been explored and some have been found to be wholly inadequate. The GEHR deliverables, which are in the public domain, contain the detailed explanations and discussions of the issues.

5.5.3 Data Transfer

The overall objective is to take data from any legacy electronic medical database system or paper based records and produce a GEHR compliant transfer file in a

standard format. The data transfer process goes through the stages shown in Figure 19.

1. Legacy System → LIF
2. LIF → GEF
3. GEF → Transfer file format (EDIFACT plus BER attachment)

Figure 19

Medical record systems are currently very different in structure and thus the conversion from the system format to LIF is different for each system. However, once the data is in LIF, the conversions to and from the GEF and EDIFACT formats, as shown in chapter 4 are identical. The process by which the data is converted between LIF, GEF and EDIFACT formats is not the subject of this chapter. The aim is to allow creation of LIF format from any proprietary system database.

5.5.4 LIF Overview

The LIF version 0.1 is a simple text format that follows GEHR structures and guidelines. The LIF is a format for data from a particular medical record system and as such, contains the information of the SOURCE (or software system) from which it originates. There may be several SOURCEs at one physical location and the process for each is the same. The LIF contains EHCRs one per patient, which contain as much or as little information as is required by a particular transfer. A certain amount of mandatory information (to identify the patient, the source of the information etc.) is required.

As well as information on the SOURCE and the EHCRs, the LIF contains information on any Health Care Facilities (HCFs), Health Care Professionals (HCPs) and other people referred to within the EHCRs. Of these, various are mandatory attributes within a GEHR structure, but not all are mandatory within the LIF.

Within a GEHR system, the recording of the HCP responsible for any medical data entered into the record and the PERSON responsible for any other data entered are both mandatory. However, in many legacy systems, this information is not stored and

it is not, therefore, mandatory in the LIF. The data cannot enter a GEHR record without an HCP taking responsibility for its inclusion.

Each EHCR is made up of a number of TRANSACTIONs (one or more). There are currently eight types of transaction defined shown in Table 7

Administrative (ADMIN)	Used to record any information which assists in the management of the patient but which is not specifically related to their health status e.g. occupation and address.
Administrative Summary (ADMIN SUMMARY)	Used to record the most up to date set of all the Administration Transactions in a record.
Contact (CONTACT)	Any information that relates to a provision of care by clinical staff in contact with a patient will be recorded within this transaction type. This kind of record entry is also known in the literature as Encounter Record or Progress Note.
Summary (SUMMARY)	Any information that is deemed to relate to the past provision of care for that patient or patient's relatives which has a relevance beyond any single transaction will be recorded in a summary transaction.
Continuing Care (CONT_CARE)	Transactions of this type are intended for information which has relevance for future transactions and relates to the ongoing clinical management of the patient. Similar to a summary transaction but relating to the future rather than the past.
Report (REPORT)	This transaction type is used for information which has a legal status outside the record. Report transactions involve communication from one responsible person to another.
Nota Bene (NOTA_BENE)	This transaction type is defined by its behaviour, as the information will be displayed whenever the record is opened. It is thus critical information relating to the patient, which the last clinician requires the next clinician to see. In many ways it is analogous to the outside cover of the paper notes.
Trigger (TRIGGER)	Any condition or information requiring action at a future date or circumstance. Trigger transactions are not dealt with in version 0.1 of the LIF as they are not fully defined in GEHR 1.0.

Table 7

Each TRANSACTION has a number of attributes that vary with the TRANSACTION type. Some of these attributes are mandatory for certain types of TRANSACTION. For example, a CONTACT must have a date of occurrence. The attributes associated with each transaction can be seen in section 2.6.2.2.2.

Within each TRANSACTION, the data is stored in Health Record Items (HRIs) and collections of HRIs (HRI_COLLECTIONs) and may be further arranged by the use of HEADINGs. It is the job of the developer of the software which will do the conversion to/from LIF (for a particular system) to see that the data is appropriately structured into HRIs, HRI_COLLECTIONs and HEADINGs. On the whole, the correct structuring is intuitive once the purpose of the HEADING and HRI_COLLECTION structures is understood.

The HRI_COLLECTION provides a mechanism for narrowing the scope of the data. The HEADING groups the data but does NOT narrow its scope. The following examples illustrate:

a) Patient's weight = 80 kg

This can be given by a single HRI within the direct scope of the enclosing TRANSACTION and thus in the direct scope of the patient.

b) The patient has a tumour that weighs 7 g

In this case, there is an HRI_COLLECTION (Tumour) under which is an HRI (Weight). The HRI (Weight) is now in the direct scope of the HRI_COLLECTION (Tumour). Hence it refers to the weight of the tumour and not the weight of the patient.

c) A Physical Examination of the Left Hand

This may be expressed as two HRIs (left index finger = stiff and left thumb = limited movement) under a HEADING (Physical Examination). Assuming that there is no other structure 'above' this other than the surrounding TRANSACTION (i.e. there are no higher level HRI_COLLECTIONs), left index finger is in the direct scope of the patient, as is left thumb. However, the two readings must be grouped together and the HEADING (Physical Examination), which indicates that these two readings belong together, does this.

Very loosely the *of* relation can be used. In b) above, it is not the weight *of* the patient that is recorded, but the weight *of* the tumour. Thus a collection is used. In a), it is the weight *of* the patient that is recorded as there is no enclosing HRI_COLLECTION to narrow the scope. In c), it is the status of the patients left hand, the two readings must be grouped together and thus the HEADING structure is used.

Further detailed discussion of structuring EHCR data into GEHR compliant form is given in [GEHR95] along with a worked example.

5.5.5 The LIF Definition

The LIF is described in Extended Bacchus Naur Form (EBNF) the full definition of the LIF can be found in Appendix E Footnotes are used to give supplementary information where appropriate.

5.5.6 Legacy Data Upgrade Path

The process to upgrade legacy data to GEHR compliance is as follows:

- Retrieve the data and decide how it should be re-structured
- Re-structure the data into a form mappable to the information model
- Create the LIF

The detail of this process is shown in the case study in section 5.6.

5.5.6.1 Retrieve the Data and Reorganise It

This, in itself, is a more demanding job the less sophisticated the system is to start with. How the stored data maps to the standard information model, identifying what is missing and determining what can safely be added to the data upon conversion has to be decided.

Examples include, identifying the HCP:

> A system which does not store the identity of the HCP seeing a patient at a consultation must have this data added only where it could be said for certain that a particular HCP saw a particular patient at a particular time, and never by guesswork. Usually the only way to do this is to go through the patient notes manually and there is rarely the resource available to do this.

Another example is obtaining the correct units for numeric data:

Many systems do not store units explicitly. The person recording the data might see:

Weight ☐ Kg

on their screen and will type a value into the box. The actual data stored may be just the value. The software may retrieve "weight" only by the physical position of the data in the database. In this example, it is safe to add the unit's kilograms to the data in the LIF, as it is clear that the intention, on entering this data, is always to use kilograms.

Systems that store minimal structure are likely to need a great deal of re-structuring. Further, the data in a legacy system will not necessarily lend itself to retrieval in a form that maps to the information model. For example, it may be difficult to gather together data for a particular episode.

5.5.6.2 Re-structure the Data into a Form Mappable onto the Information Model

Once the data has been retrieved from the legacy system and verified as correct, it can be re-structured into a form that follows the information model. Doing the restructuring at this point makes the final stage, the creation of the LIF, a straightforward process.

5.5.6.3 Create the LIF

This stage involves taking the data that had been retrieved from the system after restructuring, data item by data item and converting it to LIF.

It can be dangerous to try to incorporate all stages together where the original data is a long way from the required result as errors will be hard to spot and the integrity of the data may be compromised.

5.5.7 Conclusion

Although the data from a non-GEHR system must be structured into GEHR transactions for the purposes of the LIF, it can be reconstituted into its original form, or into any other form depending only upon the capabilities of the software system that is being used to view it. The LIF will be a simple but effective means of structuring the complete data. The richness of the underlying GEHR architecture allows that the data thus transferred may be used / viewed in the way most suitable to the specific clinical context without loss or corruption.

The GEHR architecture does not restrict systems in terms of format, structure, data set, hardware or software platform, language, term set or clinical context. Systems, both new and legacy, across Europe are known to be working towards GEHR compliance. The complete data transfer process, outlined (of which the LIF to / from proprietary system is a part) can be used as a significant step towards upgrading to GEHR compliance.

5.6 Case study

MiniClinic [Grub91] is a diabetes management system written in the early 1980s. The data within it is to be upgraded to GEHR compliance.

The original data stored is

- some administrative data about the patient
- some 'static' clinical data - stored once only for each patient e.g. height
- some chronological data e.g. weight, blood pressure and test results where a maximum of five different recordings for five dates can be stored.

Amongst data that the system does not store is date of registration, HCP data or units for numeric data. Although some chronological data is stored, it is not easy to access data by date.

The process of transferring this data to LIF is as follows:

5.6.1 Retrieve the Data and Decide how it should be Re-Structured

The data was analysed for missing items. Decisions taken included the following:

- HCP data could not be added as there was more than one GP in the practice and there had been many locums during the time the system had been in use.
- Units could be added to numeric data as it was always clear at the time of entering data which units were intended
- Coded terms could not be re-mapped. The data contained some local term sets and some free text. The code sets were checked to see if a mapping could be made to a recognised code set (e.g. ICD or Read) but there was too much potential ambiguity and it was deemed unsafe.

5.6.2 Re-Structure the Data

The data was retrieved from the system, patient by patient and output in text form. The text output was checked for consistency with the actual data in the system. The verified data was re-structured to reflect the information model. The text appeared in the following format:

Patient H Name I Title Mrs etc.... EOH I Sex 2 EOP	Transaction Administration I Registration Number 6443 I Hospital Number bb7601 I Hospital Name ... EOT	Transaction Contact 29-3-94 I Weight 76 kg H Blood Pressure I systolic BP 120 mmHg I diastolic BP 80 mmHg EOH C Tumour I Size 3 cm I Location Lower abdomen EOC EOT
etc....		

Table 8

Key:
H = Heading
C = Collection
I = Item
EOC = End of Collection
EOH = End of Heading
EOT = End of Transaction

5.6.3 Create the LIF

This text version was used to create the LIF [Grub97]. An example of the final LIF format is given below.

ntification/HCP/HCF/Source data	*Patient data*	
GACY_SOURCE acy_name="MiniClinic" acy_type=E ning_hcf=HCF1 RMSET TS1 ne="miniclinicTS1" sion="1.0" agency="HullMIG" D TERMSET P HCP1 s=uk,<"...">, "..." e="..." D HCP F HCF1 ne="..." =uk,<"...">,"..." D HCF URCE SOURCE1 ne="MiniClinic" D SOURCE	EHCR_EXTRACT SUBJECT titles="Mrs" name=... date_of_birth=... gender=F END SUBJECT ADMIN HRI<"Registration Number"> Content=6443 END HRI HRI<"Hospital number":TS1+> Content=<"bb7601”> END HRI HRI<"Hospital Name":TS1+> Content=<"..."> END HRI END ADMIN	CONTACT Dt_occurred=29-3-94 HRI<"Weight":TS1+> content=76 <"kg"> END HRI HEADING<"Blood Pressure":TS1+> HRI<"systolic BP":TS1+> Content=120<"mmHg"> END HRI HRI<"diastolic BP":TS1+> Content=80<"mmHg"> END HRI END HEADING HRI_COLLECTION<"Tumour":TS1+> HRI<"Size":TS1+> content=3 <“cm”> END HRI HRI<"Location":TS1+> content=“Lower abdomen” END HRI END HRI_COLLECTION END CONTACT END EHCR_EXTRACT *Ditto for each patient*
		END LEGACY_SOURCE

Table 9

5.7 Conclusion

Many users are locked into obsolete medical record systems and feel very frustrated when they see the power, flexibility and additional features/functionality of emerging systems. A great deal of time, effort and money has no doubt been invested in many existing legacy systems. It has been recognised that few users/system managers would be willing to contemplate upgrading to a standards compliant system unless the transition could be made relatively painless.

The LIF, the GEF and the software tools currently under development are a major step along the route towards the integration of data in existing systems with systems of the future.

Chapter 6

6 Evaluation

6.1 Chapter Overview

In order to validate the ideas described in this section, not just the exchange of information between GEHR compliant systems, but also data from legacy systems to GEHR based systems, a software package was written. This software – in essence a compiler – takes Legacy data formatted in LIF [Chapter 5] and produces the equivalent data in GEF [Chapter 4], such that it can be read by any GEF module in a functional GEHR based Health Information System.

The first sections of this chapter show the considerations that had to be taken when developing the compiler, in terms of the data that had to be added to the LIF in order to make it GEHR compliant. The chapter then goes on to explain the software development itself and how the software was designed, implemented and subsequently tested.

Finally the results are shown and conclusions drawn from these.

6.2 Considerations

The considerations presented result from:

- An initial assessment of the issues to be addressed when transferring data from an existing non-GEHR system to a GEHR based system particularly in view of the experiences of the SHINDIG project
- The import and export of GEHR Exchange Format (GEF) to/from a GEHR based system

They include consideration of how to handle the more awkward invariants in the GEHR Object Model (GOM) and propose defaults for situations where legacy systems (electronic or paper) do not contain the appropriate data.

6.2.1 EHCR Source (EHCR_Source)

If the data has been brought into a GEHR compliant system from a non-GEHR source, the Electronic Health Care Record Source (ehcr_source) will indicate both the original legacy source name and type and information regarding the tool used for converting the data to GEF. The name, Revision Identification (revision_id) and origin are that of the upgrade tool and should be added by it. The Owning Health Care Facility (owning_hcf) is that from which the legacy data originated.

The Legacy Name (legacy_name) is the name of the non-GEHR source. In the case of paper records this attribute should identify the collection of patient records being upgraded. The Legacy Type (legacy_type) will be electronic or paper

6.2.2 EHCR Extract (EHCR_Extract)

A legacy system should be able to provide a unique identifier for each record (ehcr_id); if not, it is imperative on transfer that the upgrade tool generates such an id - perhaps via an algorithm based on available data. Even with paper records, it is likely that a patient NHS number will be available.

The date and time of creation (dt_creation) is the date/time of creation of the original record (where known), not the creation of the GEF by the tool. The health care practitioner created by (hcp_created_by) refers to the HCP creating the original record (if known), not the HCP authorising the upgrade.

When creating the GEF, the upgrade tool should maintain the subject or acquired subject as they were in the original source. The receiving system will convert all of these to acquired subjects.

In the case where the legacy source has a single version of the patient's information, a single patient version (patient_version) will be generated and no acquired patients (acquired_pats).

(Note that there should be no EHCR object in the GEF.)

6.2.3 Versioned Transaction (Versioned_Trans)

The upgrade tool should generate the universal identification (uid), the date and time created (dt_created) and the GEHR version (gehr_version) used.

Access rights must be at least as tight as existing rights, unless otherwise agreed. If there are no existing access rights specified, it is recommended that they default to all HCPs + recorder + patient so that some measure of privacy is provided for.

If amendment rights are distinguished from access rights in the legacy system (electronic), then similar rules apply as with access rights - i.e. rights must be at least as tight as the existing restrictions. If amendment rights are not specified but access rights are, then the amendment rights in the GEHR system should be at least as tight

as the access rights, as there is little point in letting only some people access the record but theoretically allowing anybody to actually amend the record.

If neither amendment nor access rights are specified, the default should be all HCPs. This should be the default for paper records.

The class of transaction to be used for the legacy data should be guided by [GEHR95] and knowledge of the existing data. The problem of converting existing data to transaction form is not at discussed here. However, it should be noted that the upgrade tool should ensure that there is at least one Admin_Summary transaction to provide administration information about the patient, which could be used in part to aid the unique identification of the patient.

6.2.4 Transaction Version (Trans_Version)

By default, each transaction will have one version (revision_id = 1.0). If a legacy system is assumed to have transaction versions 1.0, then if a later transaction is received from the same legacy system but with the same assumed (i.e. 1.0) version number, then unless it can be proven that this is a later version of the same transaction, it will become a new versioned transaction (versioned_trans) with a revision of 1.0.

If the legacy system does not hold the date and time committed (dt_committed), authorising HCP (hcp_authorising) or recorder, they should be left null in the GEF and never guessed at, as there may be legal consequences. This must only happen for upgraded legacy data as any data held on a GEHR compliant system will, by virtue of the fact that the system is GEHR compliant, have this information associated with each part of the record.

6.2.5 Acquired Transaction Information (Acq_Trans_Info)

Should a legacy system hold details of transactions it had received from elsewhere, the source must be provided. The upgrade tool can provide some of these. However, there is no guarantee that the HCP Authorising acquisition (hcp_auth_acq) (HCP authorising the transaction into the legacy source) was recorded and must therefore be left null in the GEF

6.2.6 EHCR Entry (EHCR_Entry) and Sub-Classes

The upgrade tool should add the Unique Identification (uid). The nature of the Term Reference (term_ref) used for the name will depend on the type of data present and the available term-sets. The upgrade tool may choose to create local term-sets to be used, which must be sent with the GEF and the appropriate Term Set (Term_Set) and Term Set Description (TermSet_Desc) objects generated.

The upgrade tool should ensure that empty HRI Collections are not created.

6.2.7 Regestration Agency (Reg_Agency)

Where the upgrade tool has generated local term sets, the source of the Local registering Agency (local_reg_agency) should be the same as the legacy source for the EHCR. The Registering Agency (reg_agency) name should indicate, unambiguously, the body responsible for the generation with respect to that source.

6.2.8 Units

In converting from quantities where the units are ambiguous or absent, then the unit should be recorded as text. If no unit term set is available, one must be created. The upgrade tool can automatically generate a unit term set as it parses the LIF. The term set can then be transferred with the LIF if required by the receiving system.

However, if all the details of the units are not sent and the receiving system does not have a copy of the relevant unit term set it will not be possible to accurately interpret what is received. All that can be done is to display what is received *as is*.

6.2.9 Transfer of codes from a Legacy System

If the original system has only its own coded terms then in converting this to GEF data for transfer to a GEHR system there are two options:

- Use of own codes and becoming a local registering agency
- Convert these codes into a more widely known code set e.g. read or ICD9

6.2.9.1 Use of Own Codes

This faithfully represents what was in the original. The disadvantage of this is when the need arises to carry out a search, terms from all different sets of codes from many

different term sets would have to be examined in order to find the correct item that is being searched.

Having transferred this code set to any recipient the code set will continually have to be maintained by a local registering agency. Becoming a local registering agency obviously has consequences associated with it and cannot be done light heartedly. It may be that an overseeing body would have to be set up in order to manage these local registering agencies.

If the original legacy system only used free text for item names then these have to be converted into terms irrespective of whether they are local terms or terms from a code set that is more widely used. The GOM does not allow free text as item names for some very good reasons, which are explained in [GEHR95]. One of these reasons being the problems that are inherent when analysing information for epidemiological studies.

6.2.9.2 Convert to Another Code Set

If an attempt is made to convert local names to codes from a recognised coding scheme the fact that the codes were translated from another, maybe lesser known, term set should always be recorded. It may also be necessary to transfer these terms if the system does not know about these terms or if an audit trail of where to find a definition of the terms is not provided.

The issue of cross mapping of term sets is outside the remit of this work.

6.3 The Software Design

In order to evaluate the ideas that have been suggested, a program was written to test the theory (the software is available from the author). The rest of this chapter presents the design and implementation of the software program that was known as a Compiler for Legacy Information Format (CLIF).

The language of the LIF is formally defined in *Extended Bacchus Naur Form* (EBNF). The full definition of this language can be found in Appendix E, which contains the syntax and semantics of the language as well as the Grammar. The phases of the compilation process can be seen in figure 20.

The compiler analyses the input, which in the case of the CLIF is medical data in LIF. It then outputs the results of the analysis as an output production. The output of the

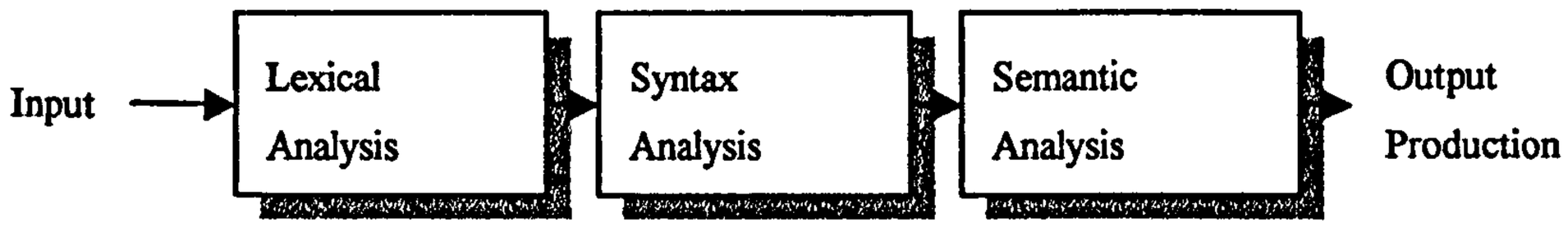


Figure 20

CLIF is the corresponding medical data in GEF.

The CLIF software consists of two main parts. The first of these is the Lexical Analyser with the second part incorporating both the Syntax Analysis and Semantic Checking. The CLIF is therefore said to be a two-pass compiler.

The path through the language can be seen in the state chart diagram, Appendix F.

6.3.1 First Pass - Lexical Analyser

The lexical Analyser parses the input LIF file, to check the grouping together of strings of characters denoting identifiers, constants or language words into single tokens, which are in line with the formalised EBNF language. It then outputs this into a token file that holds the references to the Identifier, Date, String, Symbol or Value that are held in different files.

After defining the language a symbol table was created. This holds all the reserved symbols that appear in the language. These are defined in the file symbol.txt and are:

" < > + ( ) : - % ^ = , . --

The next stage was to define the reserved keywords used in the language. These were stored in a file called keyword.txt, and are presented below:

```
Legacy_source
legacy_name
legacy_type
owning_hcf
TERMSET
DEFAULT
UNITS
HCP
END
STAFF_MEMBER
TOOL
END TOOL
```

```
NON_PATIENT
HCF
SOURCE
PAPER
ELECTRONIC
code
name
revision
reg_agency
reg_agency_source
EHCR_EXTRACT
regs
```

profession
grade
position
tool_type
addresses
contact_nrs
net_addrs
photo
dt_occurred
type_of_hcf
reg
revison_id
origin
ehcr_id
dt_creation
hcp_created_by
SUBJECT
hcp_authorising
hcp_legally_resp
recorder
dt_committed
ADMIN
ADMIN_SUMMARY
period
POSSTR
TITLES
letters
date_of_birth
gender
addr_lines
postcode
valid_from
invalid_from
ADDRESS
access_rights
amend_rights
dt_occured
contact_with
CONTACT
NOT
SUMMARY
REPORT
in_reply_to
CONT_CARE
NOTA_BENE
uk
UKeng
FEMALE
MALE
UNKNOWN
HEADING
emphasis
FROM
INC
TO
derived
HRI
ct_comment
dt_comment
certainty
ct_emphasis
content
HRI_COLLECTION
MEDIUM
HIGH
links
info_provider
TRUE
FALSE
OF
PER
ON
IS
BY
AS
E
M
F
SF
DP

6.3.1.1 Algorithm

The algorithm for the Lexical Analysis is as follows:

```
Open Input file
Loop
```

```
Read first character from input file
Define what type character is
If Symbol then
        Store as symbol
Elseif an alphabetic character
        Read next character
        If alphabetic character or numeric
                Build up complete word
        Compare the string to the reserved keyword list
                If it is a keyword
                        Append a token in token file of type Keyword, reference the
appropitate keyword in the keyfile
                Elseif not a keyword
store as an identifier
Enter reference to the identifier in the token file
Endif
Endif
Elseif quotation mark
                Build up string until next quotation mark
                Save as a string in the string file
                Enter reference to the string in the token file
        Elseif a numeric
                Build up number
                If number is a date
                        Store date in date file
                        Enter reference to date in token file
                Elseif integer
                        Store as an integer in the value file
                        Enter reference to integer in token file
                Elseif real
                        Store as a real in the value file
                        Enter reference to real in the value file
                Endif
        Elseif a hyphen
                Check the next char
                If a hyphen
                        Ignore the line as this is a comment
                Elseif a numeric
                        Build up number
                        If an integer
                                Store as an integer in the value file
Enter reference to integer in token file
                        Elseif real
                                Store as a real in the value file
Enter reference to real in token file
                        Endif
                Endif
        Elseif a space
```

```
            ignore
        Endif
Endif
Until End of file
```

Table 10

6.3.1.2 Input File

An example input file of LIF is shown below:

```
LEGACY_SOURCE
legacy_name="Miniclinic"
 legacy_type=E
 owning_hcf=HCF1
TERMSET TS1
  code="0123"
  name="miniclinicTS1"
  revision="1.0"
  reg_agency="Termset agency"
END TERMSET
HCP HCP1
  regs=uk,<"GarysReg">,"STR
Gary's registration"
  name="Gary"
END HCP
HCF HCF1
  name="Manor Park"
reg=uk,<"ManorParkReg">,"STR
Manor Park's registration"
END HCF
SOURCE SOURCE1
  name="MiniClinic"
END SOURCE
EHCR_EXTRACT
 SUBJECT
   titles="Mr"
name="Kevin","Gordon","Neville"
  date_of_birth=28-12-1956
  gender=M
 END SUBJECT
 ADMIN
  HRI<"MicroDoc    Registration
Number">
   content=1939
  END HRI
  HRI<"Hospital Number":TS1+>
   content=<"433488">
  END HRI
  HRI<"Hospital Name":TS1+>
   content=<"Northern    General
Hospital">
  END HRI
  HRI<"Occupation":TS1+>
   content=<"Unemployed">
  END HRI
 END ADMIN
 CONTACT
  dt_occurred=18-1-94
  HRI<"Hypoglycaemia":TS1+>
   content=<"No">
  END HRI
  HRI<"Therapy Change":TS1+>
   content=<"No">
  END HRI
  HRI<"Weight":TS1+>
   content=131.2<"kg">
  END HRI
  HEADING<"Blood
Pressure":TS1+>
   HRI<"systolic BP":TS1+>
    content=154<"mmol">.<"1">^-1
   END HRI
   HEADING<"Blood
Pressure":TS1+>
    HRI<"diastolic BP":TS1+>
     content=96<"mmHg">
    END HRI
   END HEADING
  END HEADING
 END CONTACT
END EHCR_EXTRACT
END          LEGACY_SOURCE
```

6.3.1.3 Output Files

The Lexical Analyser parses the LIF input file and tokenises it to create a master token file. The input file is broken down into types. For each type a reference identifying the type either; Symbol, Keyword, Identifier, Value or Date and the location of the data is entered into the token file.

Several intermediate output files are created during the Lexical Analysis process these files are used as input files for the second parse of the compiler, the semantic checking. The files are date, string, value and identifier. Along with these files, two files that are held as reference files are the keyword and symbol files. The Keyword file contains the strings that are reserved in the LIF language, whereas the symbol file contains the reserved symbols in the language.

6.3.1.4 Date File

Any date that occurs in the input data file, when parsed by the Lexical Analysis, is stored in the date file. An entry giving the position and indicating it is a date is entered into the master token file. The date file is a random access file.

6.3.1.5 String Files

The string file holds all the strings that occur in the input data file. Strings are appended to this file and an entry is placed in the string index file indicating the position that the string occurs in the string file and the length of the string. An entry is appended to the master token file giving the type of entry, in this case a string, and the reference position in the string index file to obtain the index for the string.

6.3.1.6 Value File

The value file holds all the numeric data that occurs in the input data file. Each entry in the value file indicates whether the numeric is of type real or integer. When the Lexical Analyser recognises numeric data it appends a token in the token file

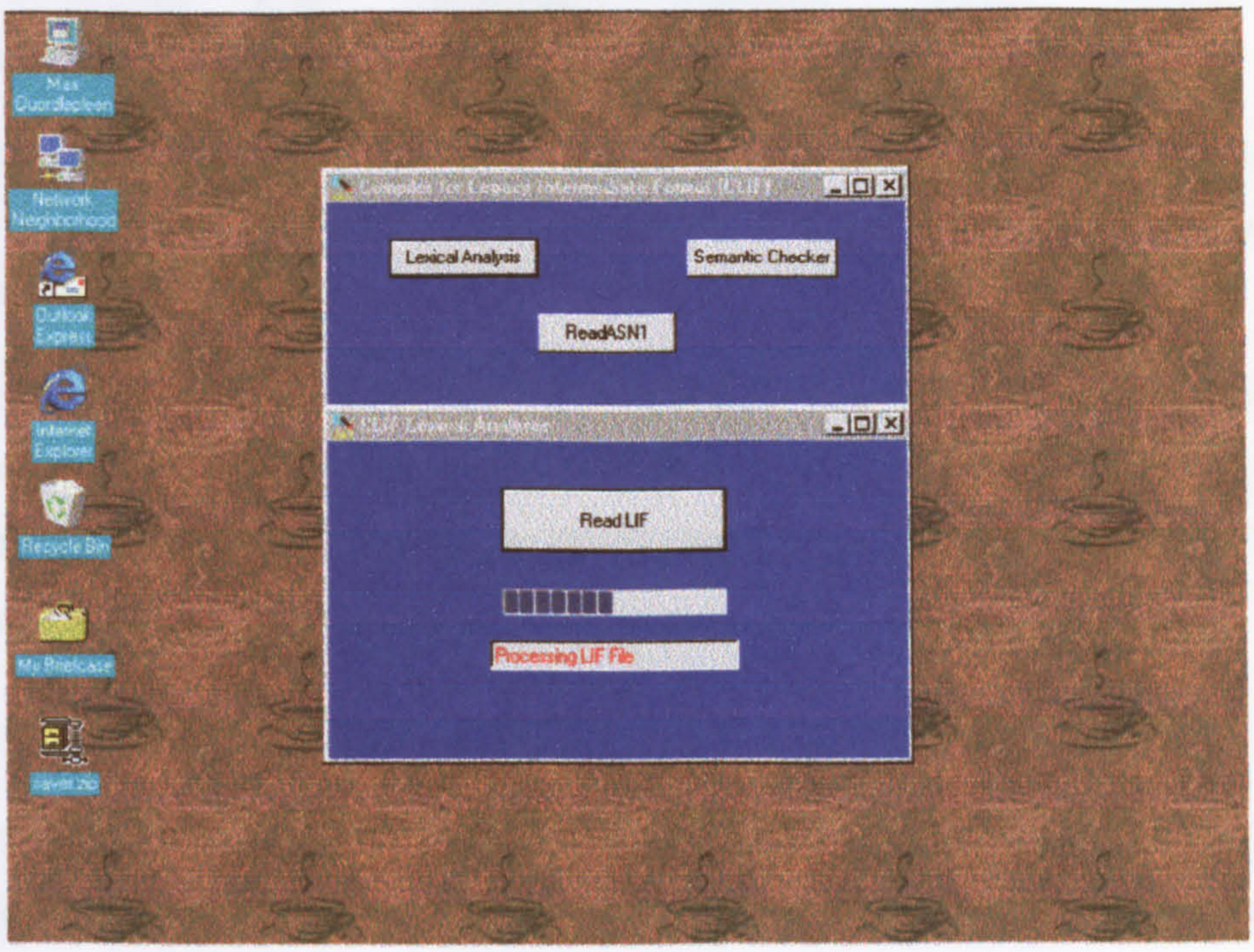


Figure 21

indicating that the data is numeric and its position within the value file which can be randomly accessed.

6.3.1.7 Identifier File

The identifier file holds all the input data that identifies things such as termset, Health Care Facility, Health Care Practitioner etc. Identifiers are recognised as anything that is not in the keyword file or not in quotes. When an identifier is recognised an entry is appended to the token file indicating that the token is an identifier. The identifier itself is placed into the identifier file and an index entry to it is placed in the identifier index file.

6.3.1.8 Token File

The token file identifies the sequence in which the tokens are to be read by the next pass of the compiler.

6.3.2 Second Pass – Syntax and Semantic Checker

The second part of the compiler incorporates both the syntax checker and the semantic checker phases. This part of the program takes for its input the token file that was output by the lexical analyser.

The syntax analysis is defined as being *'the rules defining the legal sequences of symbolic elements in a language'* [Comp96], with the language in this case being the LIF.

The role of the semantic analyser is to determine whether the input file (the token file) grouping, and associated meaning, of symbols is legal. Associated with this part of the compiler is the output production. As the structure of the input is verified the relevant translation into GEF takes place and the output productions are created.

6.3.2.1 The State Chart

The formalised version of the LIF, in EBNF format, was used to create a state chart defining the legal paths through the LIF. The top-level state chart diagram can be seen in figure 22.

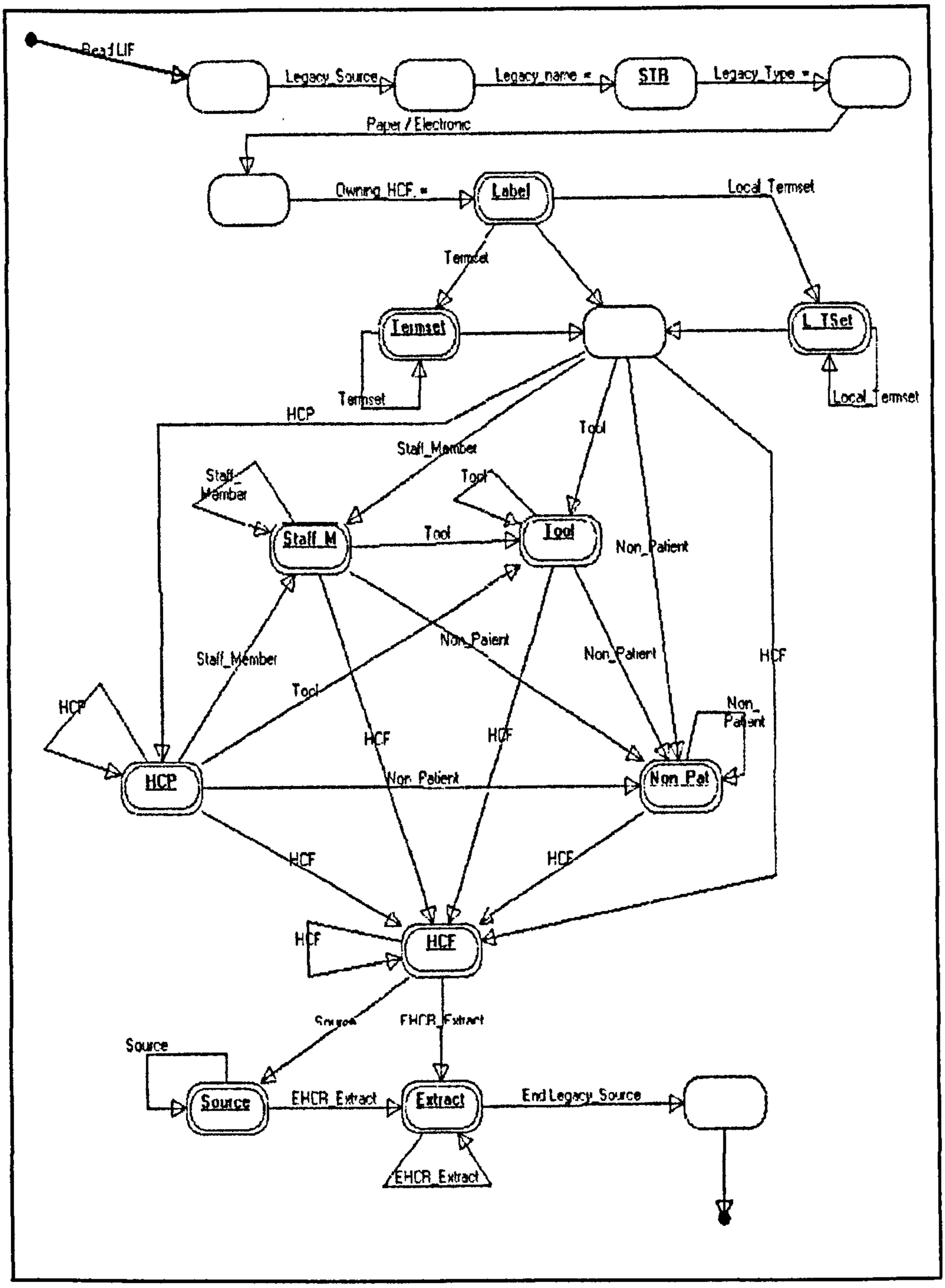


Figure 22

The full state chart can be seen in Appendix F. The arrows in the state chart diagram show the sequences of data that have to be parsed in order to be legal. At each state, represented by the oval shapes, there may be a sub-state. If this is the case a smaller oval shape appears inside the state. The sub-states in turn define the conditions that have to be met in the language for a successful parse through the LIF. Some of the states may in turn have there own sub-states, and so on.

The syntax checker reads the token file and decides whether the token is legal for the stage it is at in the language. If it is not then an error message is returned indicating that the syntax is incorrect. If it is suitable then the next token is read. At the same time as checking the syntax the program will output the relevant parts of the LIF in GEF, in ASN.1. The checker will also add extra relevant parts that are required in the GEF.

The CLIF syntax checker has a grammar that is said to be *LL(2)* this means that by looking ahead no more that 2 tokens from any position in the token file it can work out which production to apply at any stage. The *LL* part means that the input file is read from left to right. [Wats89]

The result of the syntax checker is an output file containing GEF, which can be read by a GEHR compliant system, upon receipt and incorporated into the receiving applications database. The example in Appendix G shows the output from the CLIF in GEF, for the LIF input file as described in section 4.2.1.2.

6.3.2.2 ASN.1 Output

If an input string is found to have a legal syntax and a correct meaning, correct semantics, then the program calculates the output production. The output productions are calculated and output in the form of an ASN.1 string. The program calculates each part of the ASN.1 as shown in section 4.5.1. There are three main parts to it the identifier, the length and the value that it contains which in turn if the value is constructed will hold other types and values.

If the syntax and semantics are correct then the token that has been read is encoded into ASN.1 BER in the following way. Firstly the identifier is calculated:

```
t1 = enc_tag(Class Type, Primitive or Constructed Form, Tag Number)
```

The algorithm for encoding the tag is shown below:

```
Function enc_tag(ByVal class As Integer, ByVal codeform As Integer, ByVal
tagnumber As Integer) As String

Dim s As String
s = ""
If tagnumber < 31 Then
```

```
  s = Chr(class + codeform + tagnumber)
Else
  'Long form
  s = ""
  While tagnumber >= 128
    s = Chr(128 + tagnumber Mod 128) & s
    tagnumber = tagnumber \ 128
  Wend
  s = Chr$(128 + tagnumber) & s
  Mid$(s, Len(s), 1) = Chr$(Asc(Right$(s, 1)) - 128)
  s = Chr(class + codeform + 31) & s
End If
enc_tag = s
End Function
```

The next step is to encode the actual data that is gleaned from the token file, this in *itself can go through multiple processes and be encoded many times before the base* type is encoded. This can be seen as:

```
t2 = data
```

Where data can contain many encodings itself.

Next the length of the tag is calculated:

```
tag_len = Len(t2)
```

This is finally all put together:

```
t = t & t1 & enc_length(tag_len) & t2
```

and written to the output file.

Figure 23 shows the second pass of the CLIF specification.

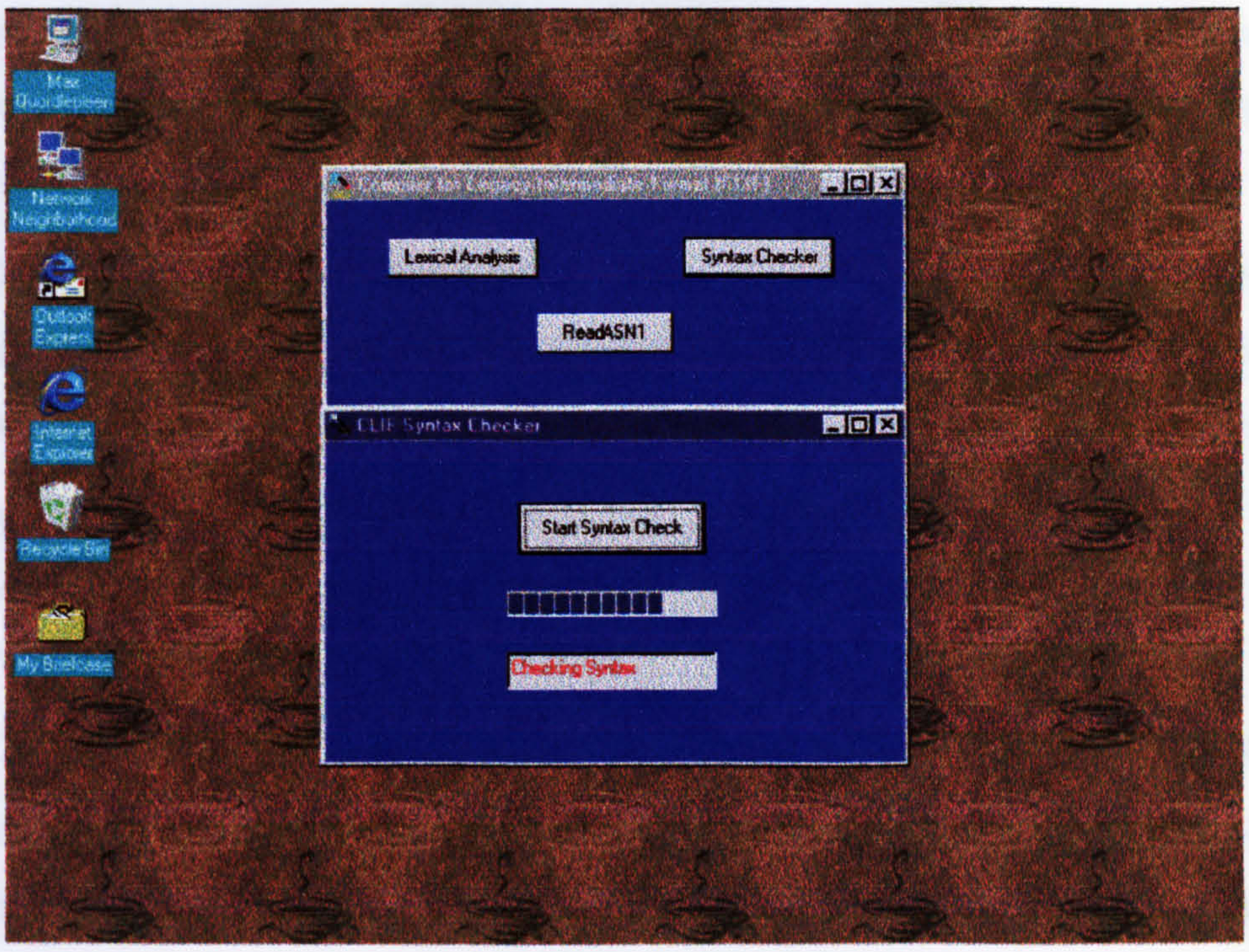


Figure 23

The CLIF software was written in Visual Basic.

6.4 Test

The testing of the output file from the CLIF software was subject to two methods of testing:

- The formal method
- The integrated method

6.4.1 The Formal Method

The CLIF software includes a module that is a GEF reader. This takes as its input the ASN.1 GEF file that is created by the other modules of the CLIF software. It then reads the file and outputs in a readable format the contents of the file to the screen. Although the output is not displayed on the screen in the same way as you would expect to see a fully GEHR compliant system display it, the reader quickly verifies that the GEF file is valid and can be read.

6.4.2 The Integrated Method

This method of testing the output file had the end goal of transferring the legacy data to a prototype GEHR system that had a module that enabled it to read GEF and incorporate this into the systems patient database.

The prototype GEHR system that was used to read and store the legacy data prepared by the CLIF software is known as PRISM (PRototype Information System for Medicine) and was a result of the SHINDIG project [Dixo99].

The CLIF software prepared the data for input into the PRISM system in the same way the data was prepared for the formal method of testing. This was read into PRISM using the GEF module, which stored the contents of the file in the system.

The specification of the computer on which the CLIF software was executed was a Pentium II 200Mhz machine. Taking a LIF input file containing the medical details for 10 patients it produces a GEF output file in 1 minute 34 seconds

When the CLIF software is producing the GEF output file the semantic checker checks that the next token that is parsed is legal in terms of the LIF language. It does not check the content of the LIF to check, for instance whether the units for a result are the correct units.

The weakness of the CLIF software is that the input file can be relatively small e.g. 24kb but the software will produce an output file which is much larger approximately 508kb. This is because the output file contains more contextual information than the legacy data. This is a weakness, as the file will take some time to transfer over conventional communication systems. However, it is envisaged that this will become less of a problem as time goes by and the bandwidth available for communication between machines increases.

6.5 Conclusion

The implementation of the CLIF, and the subsequent testing of the software, showed that the legacy data could be output into GEF in the form of ASN.1 and transferred from a legacy system to a GEHR compliant system.

It is important to recognise that as legacy data stands it is not future proof. However if it can be upgraded to be compliant with future standards for electronic healthcare records then it will become future proof. This chapter has thus demonstrated an important concept.

Chapter 7

7 Conclusion

7.1 Overview

The original aim of this study was to investigate current methods for the transfer of computerised medical data. There are several reasons why it is important that a generic method for the transfer of medical data can be defined, one of the most important reasons is identified by Grubb [Grub91]:

"...to use information technology to the full, information systems must interact"

In chapter 2 it was shown that current adopted methods for the transfer of medical data in this country are inadequate. Reasons for this are based on the complexity of the data itself, as well as the information that needs to be transferred to make the data comprehensive.

In chapter 3 other alternative methods were investigated, as well as new emerging techniques. Whilst some of these methods are thought to be interesting for further consideration at the time of investigation the technology was not mature enough for consideration.

It was recognised that in order for data to be portable it must be based on a standard architecture. One such emerging European standard, the GEHR object model, was presented in chapter 4. With such a standard as a basis an exchange format could be defined. The major innovative part of the work undertaken was also presented in chapter 4 this being the GEHR Exchange Format.

In chapter 5 the area of existing, or legacy systems was investigated. The second major part of innovative work was described in this section, the Legacy Intermediate Format. This allowed data in paper based or non-standard conformant systems to be transferred to standard compliant systems.

An evaluation of the LIF was then undertaken. A compiler was written to take data from the non-standard systems once it had been put into LIF. The compiler would take LIF as its input and output GEF adding all the additional information that was needed. This work was presented in chapter 6.

7.2 What has Been Achieved

Two main things have been achieved during the work in the area of data transfer of medical data. Firstly an exchange format was devised which was based on the GEHR

architecture. Secondly work on upgrading from legacy systems was undertaken culminating with the definition of a Legacy Intermediate Format that allowed data to be taken out of existing non-compliant standard systems, or paper based systems, and transferred into standard compliant systems.

The work on the GEHR Exchange Format highlighted many points of interest. It showed that the transfer of medical data could indeed be carried out in a generic way. The program that was written to automate the transfer highlighted errors in the original GEF, and showed what needed to be changed in order to carry out data transfer. This helped to refine the GEF.

One of the main points of interest that resulted from this work was the actual method used to express the GEF, the ASN.1. It was felt that the actual encoding mechanism is irrelevant as long as it is appropriate to use for data exchange. What is really important is maintaining the structure of the data as it is being transferred, including all the extra contextual information that is vital to accurate interpretation. Another exchange format that is emerging and may be used is XML.

It was felt that the Legacy Intermediate Format was a useful thing to define as this is going to become an area of great interest as standards are introduced and software companies are going to have to think about how to transfer the data held in their existing systems into new systems that are being created. Again this work highlighted some weaknesses in the GEF and served as a good method for tightening up the definition of the GEF. The work that was carried out would simplify the work to be carried out when upgrading existing data. Again the use of ASN.1 in the legacy upgrade was a precursor to formats such as XML. With the definition of standards it is thought that XML now takes the place of ASN.1 and could be used to help in the definition of legacy upgrades.

7.3 How the work Contributes to the Knowledge Base

The work that has been carried out during the period of research into the transfer of data has contributed to the knowledge base in many ways. Firstly the analysis of methods of data transfer at the beginning of the work has served to highlight major weaknesses in methods that were available at the time.

On a more positive note work has been carried out that has influenced the thinking of major European projects such as GEHR and contributed to the works of the follow on

project EHCR_SupA. This has in turn fed into standards making organisations such as CEN and ISO. The work from GEHR, which included the first definition of the GEF, has been presented directly to de facto standards organisations such as Corba™.

7.4 Future Work

It is recognised that in the computing industry there is a technological revolution every six years [Coch96], which is incredible when considering that in most industries revolutions take place once in a lifetime. During these six years the technology we use moves on at a staggering pace. It is with this in mind that the conclusions in this section have been drawn.

The differences between messaging and interoperability have been highlighted. The work presented in this thesis has concentrated on the messaging side of data exchange. Interoperability is coming more to the fore and it is an area of great interest that should be looked into. That is not to say that messaging does not still have a place. However in order to take data exchange to its logical next step interoperability is needed. This is where emerging formats such as XML may come to the fore as they can be used in an interoperable environment. It must be remembered however that the most important point is that the data must be structured and these structures must be based on standards.

Paper based systems will need to be upgraded to electronic systems. Whilst the LIF was one way of coping with this transition there may be other ways that will help this interface.

The actual physical transfer mechanisms will also need to be addressed now that mobile communications are available at competitive costs.

Work on the human angle will also have to take place, in the form of appropriate training, so that people will feel comfortable when using the technology presented to them and not to be scared off by it. This is one area that will hinder the introduction of data exchange technologies if not considered appropriately.

However, the main area of work should be addressing standards for the storage and transfer of healthcare data. Without standards the days of being able to transfer comprehensive medical data in a generic way will never be a reality.

7.5 Conclusion

The main driving force behind the work presented in this thesis was *"to improve the effectiveness of healthcare through the effective use of Information Technology"*. Again Grubb highlights that [Grub91]:

"Much valuable information is lost between incompatible information systems to the detriment of healthcare"

It is hoped with the introduction of standards based architectures for healthcare systems that exchange formats such as the one presented and the definition of formats to upgrade to standards based systems will help to rectify this situation and indeed improve the effectiveness of healthcare.

Appendix A –EBES Letter

Letter from the European Board for EDI Standardisation (EBES). This letter explains why the proposed 'Snn' solution to avoid segment collision was rejected. See section 2.5.2.1

Mr Stuart Campbell
Technical Officer, EBES Secretariat
Tel: +322 5500874 Fax: +322 5500909
Internet: campbell@cenclobel.be
X400: C=be, A=rtt, P=cenclobel, S=campbell

EBES Secretariat
Tel: +322 5500889 Fax: +322 5500909
Internet: ebes@cenclobel.be
X400: C=be, A=rtt, P=cenclobel, S=ebes

EBES
European Board for EDI Standardization
SECRETARIAT
European Board for EDI Standardization

Mr Jeremy Ellis
The University of Hull
Cottingham Road
HULL HU6 7RX
UNITED KINGDOM

File : C:\DOCUMENT\SC517ELL.DOC
Date : 96/05/17 15:50

Dear Jeremy:

The Snn proposal was rejected by the PA, AZ and AS regions broadly on the same basis, i.e. 'it should be a syntactical solution', and to formally reject Snn, they 'took advantage' of a technical assessment rule on duplication of functional since the function of S01=S02=S03=etc.

The position should have been negotiated at the JRT JTAG but, due to the mutually exclusive views and the minority position of EBES, little discussion eventually took place. EBES's concern of this fact is expressed in the JTAG minutes.

The subject of collision will be reviewed at an interim JTAG meeting just before the Helsinki JRT. JM 11 also raised an accepted resolution which urged further anti-collision work should take place.

Regards

S. Campbell

Stuart Campbell

EBES Secretariat

European Board for EDI Standardization (EBES)
c/o CEN
Rue de Stassart, 36 B-1050 Bruxelles
Tel : +32 2 550 08 11 Fax : +32 2 550 08 19

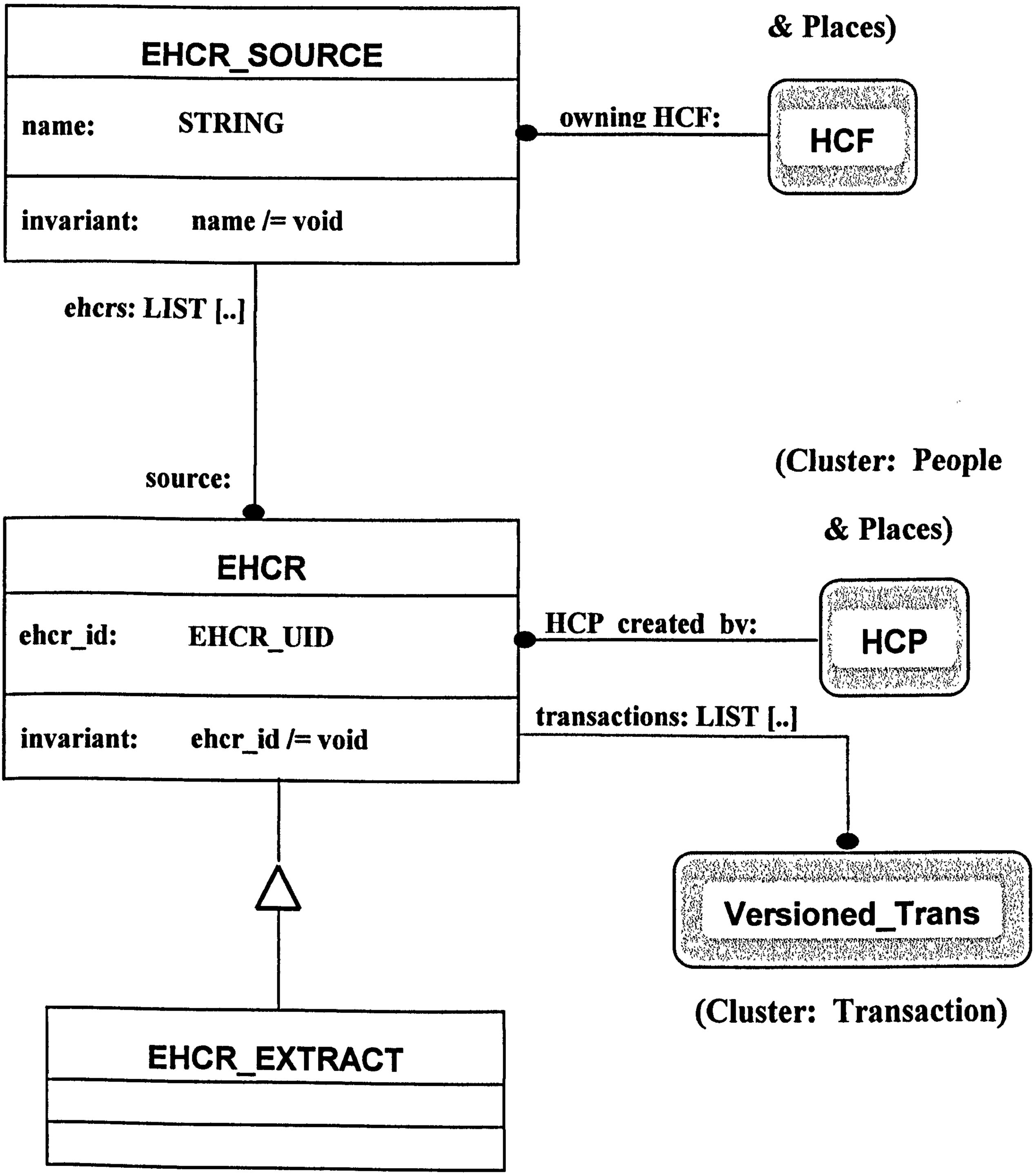
Appendix B – GEHR Object Model v1.0

GEHR Object Module Version 1.0. The following diagrams show the GEHR architecture, see section 2.6.1. The diagrammatic representation of the model is based on the "Rumbaugh" methodology, together with concepts from the Eiffel language and the BON notation.

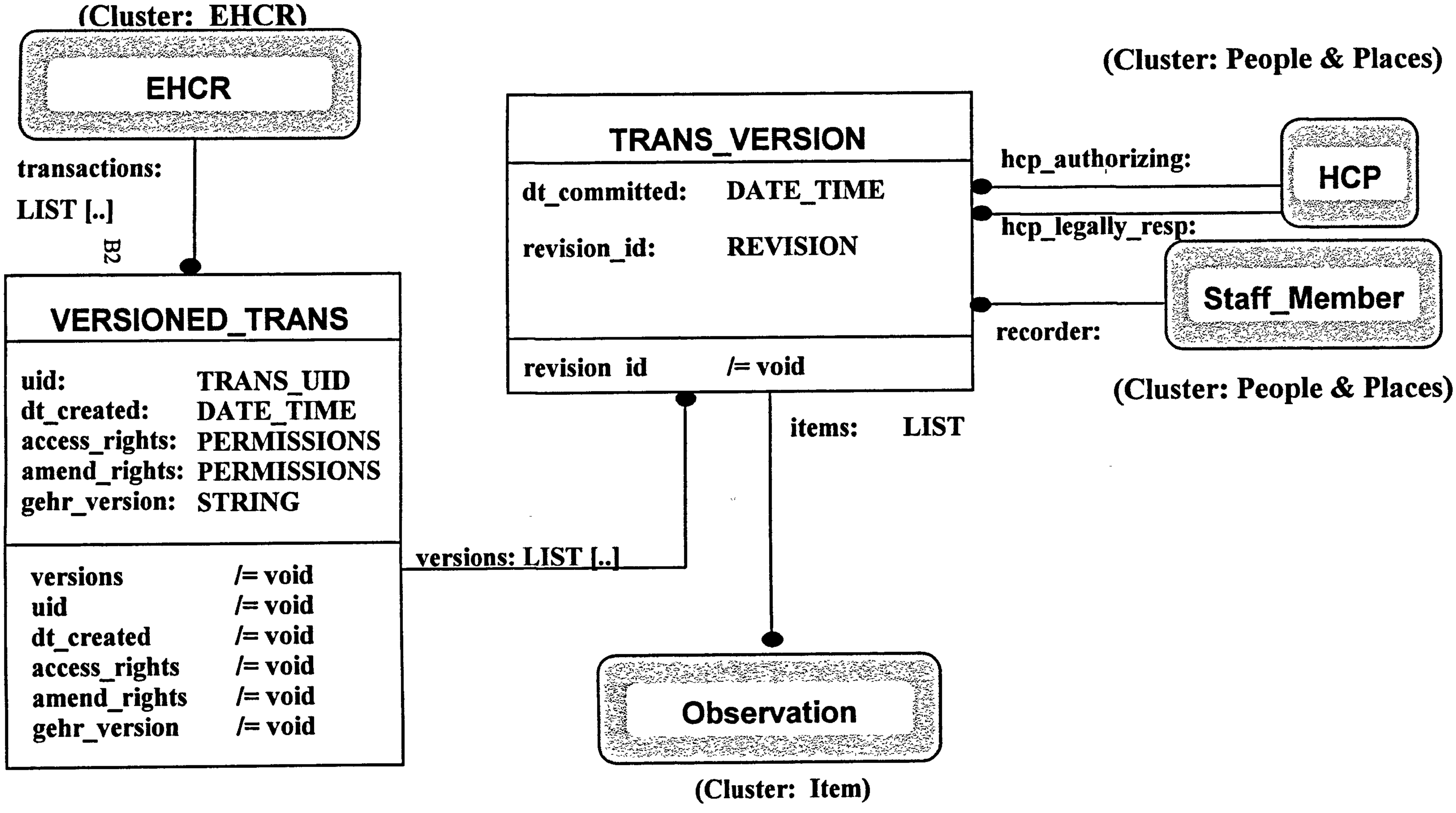
TEXT BOUND INTO

THE SPINE

EHCR Cluster

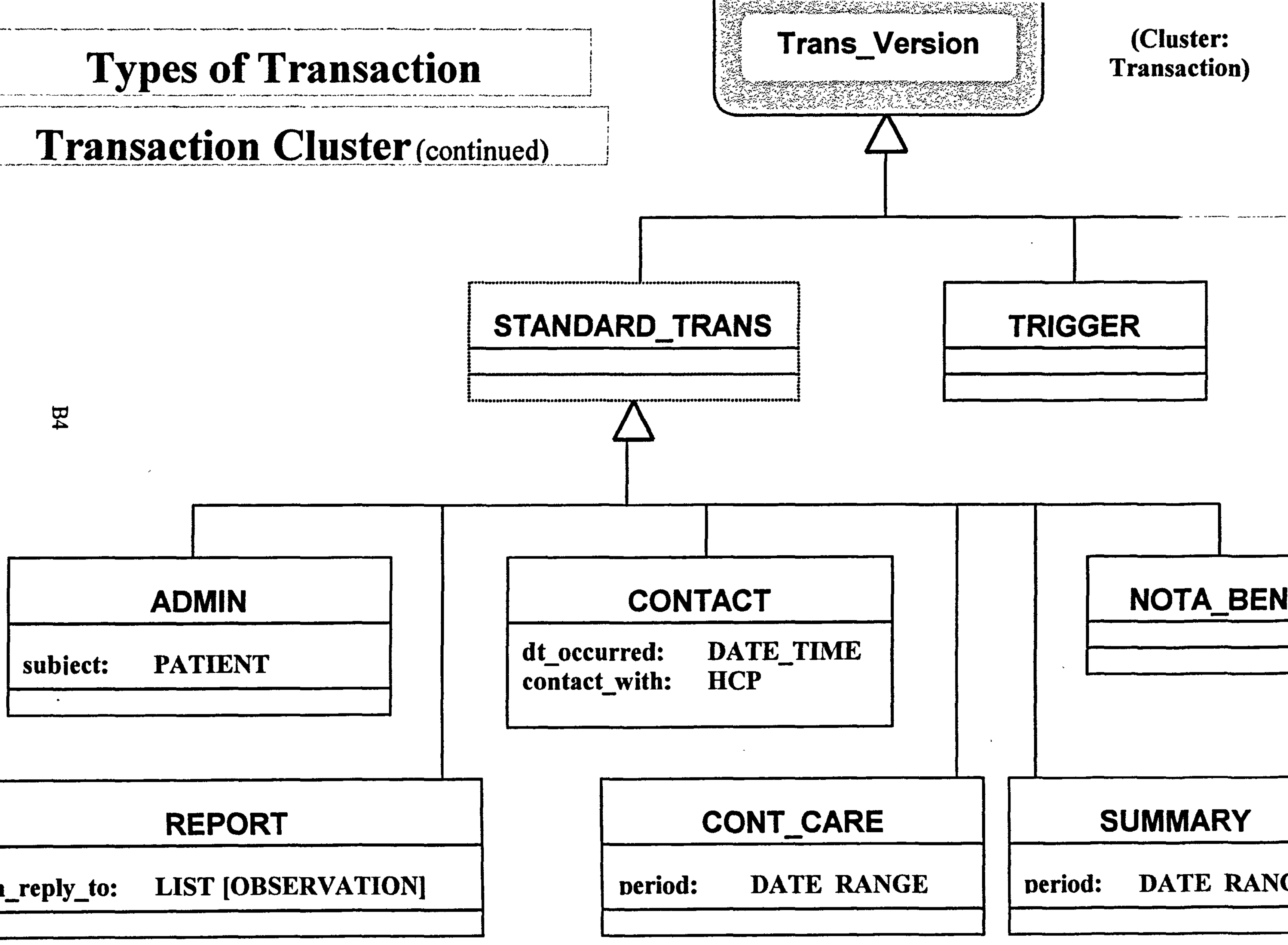


Transaction Cluster



Types of Transaction

Transaction Cluster (continued)



Item Cluster

(Cluster: People & Places)

Staff_Member

recorder:

(Cluster: Transaction)

Trans_Version

EHCR_ENTRY

name:	TERM_REF
emphasis:	EMPH_LEVEL
uid:	ENTRY_UID

name /= void

Person

info_provider:

HEADING

parent:

annotated_by:

SUBSET [..]

OBSERVATION

links:	ARRAY [GEHR_UID]
access_rights:	PERMISSIONS
cx_comment:	MULTI_TEXT

in_reply_to:

LIST [..]

uid /= void

members:

HRI

ct_comment:	MULTI_TEXT
dt_observed:	DATE_TIME
certainty:	PLAIN_TEXT
is_derived:	BOOLEAN
ct_emphasis:	EMPH_LEVEL

content

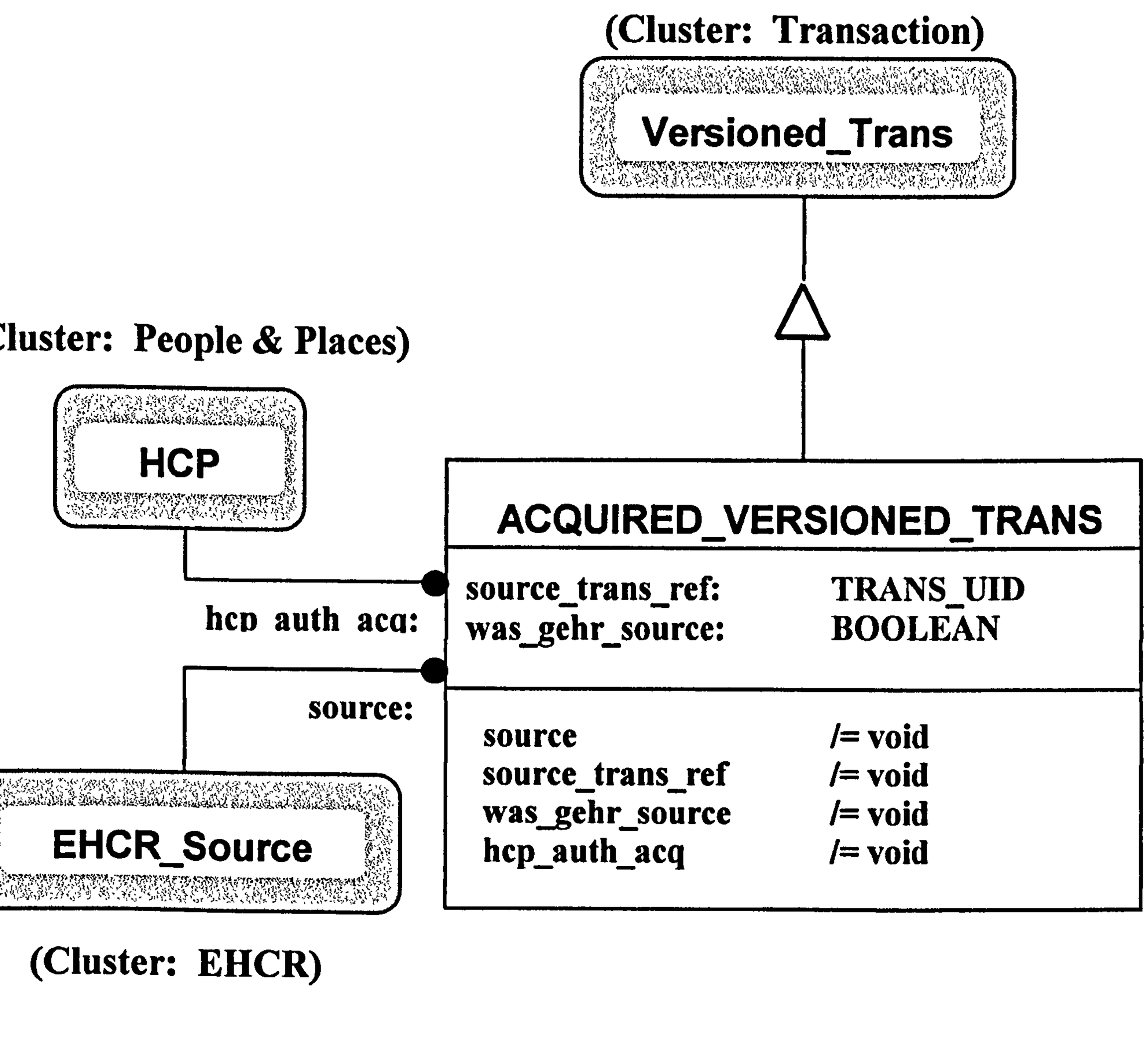
EHCR_Info

HRI_COLLECTION

not members.empty

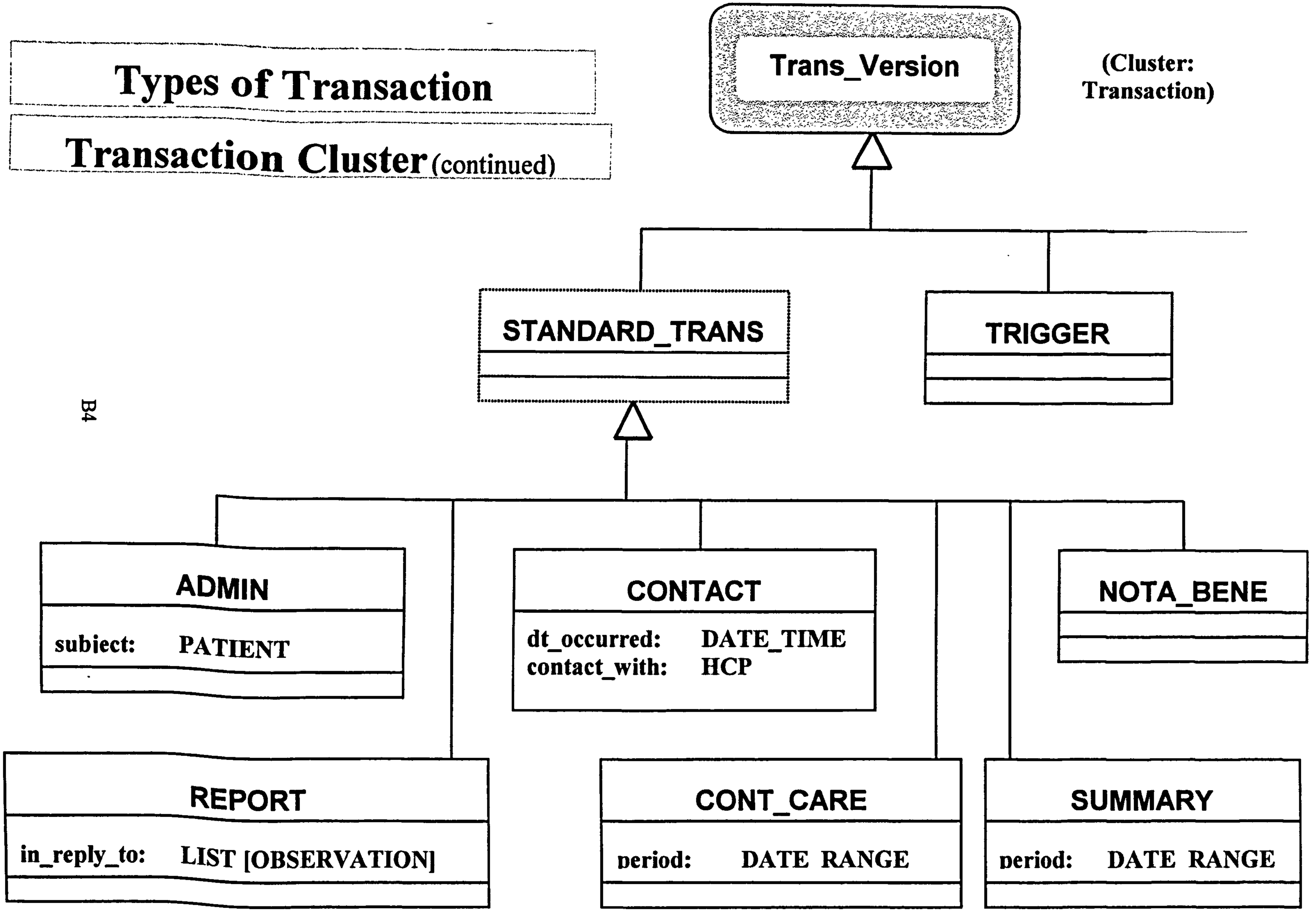
(Cluster: EHCR_Info)

Transaction Cluster (continued)

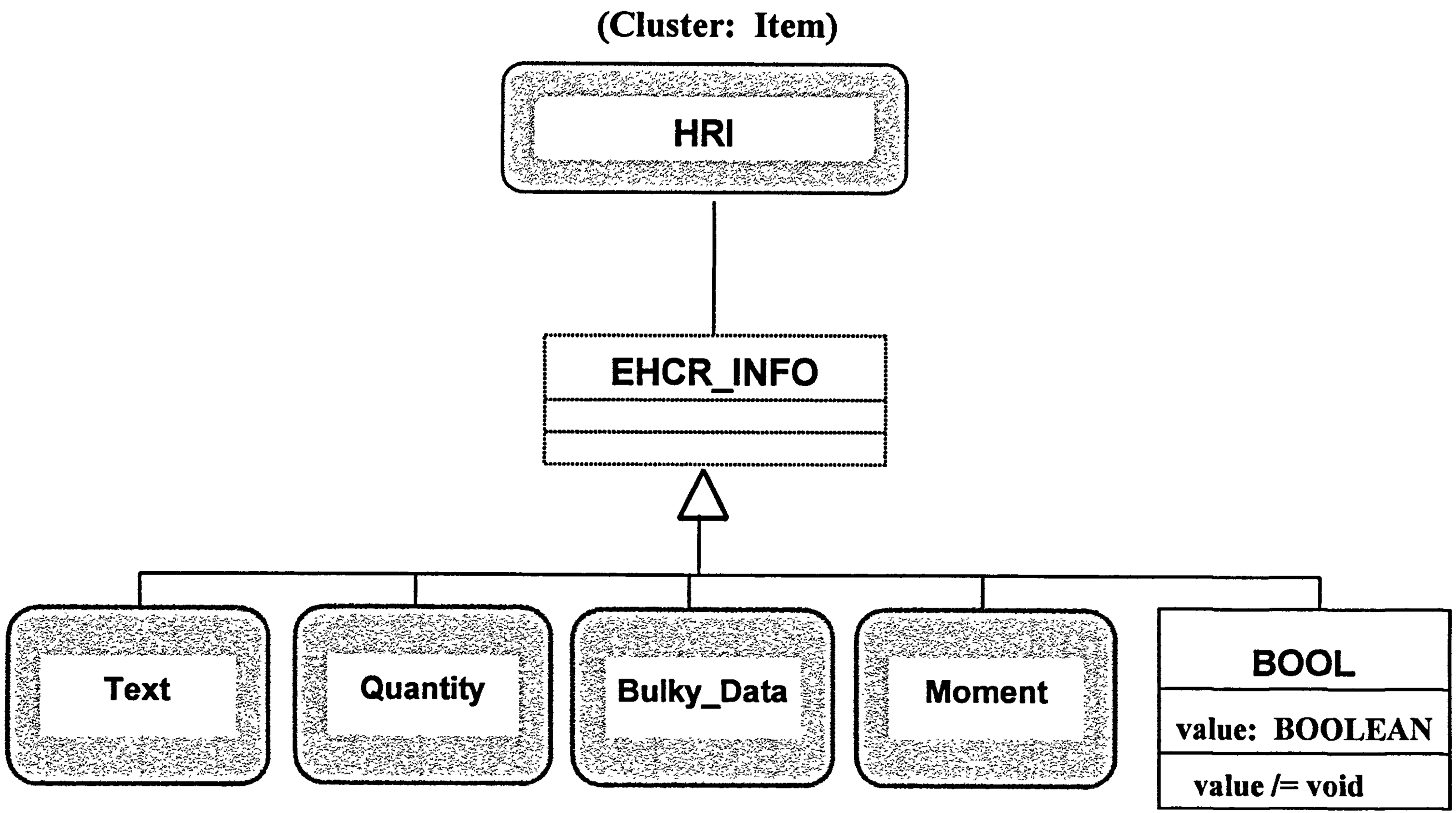


Types of Transaction

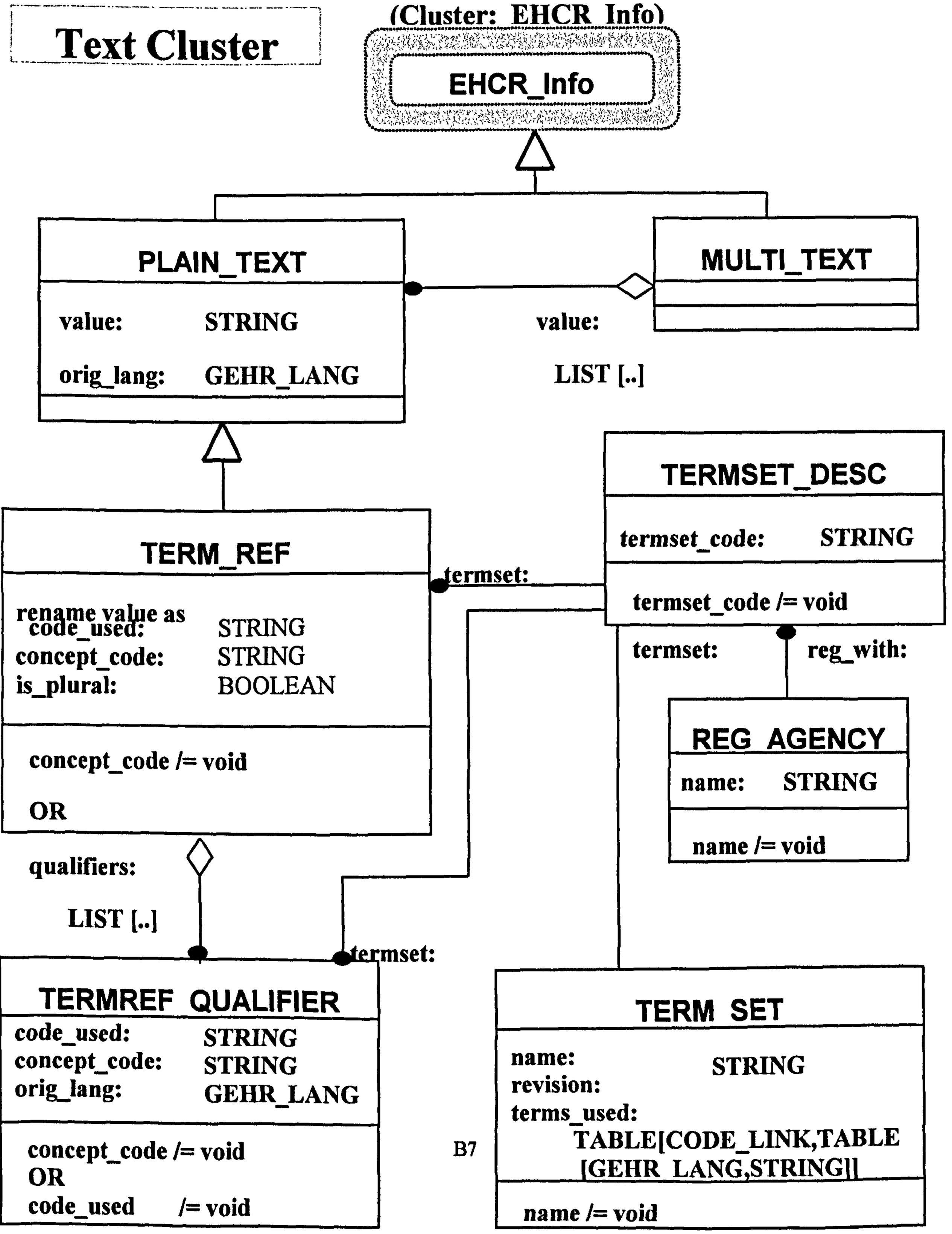
Transaction Cluster (continued)



EHCR Info Cluster

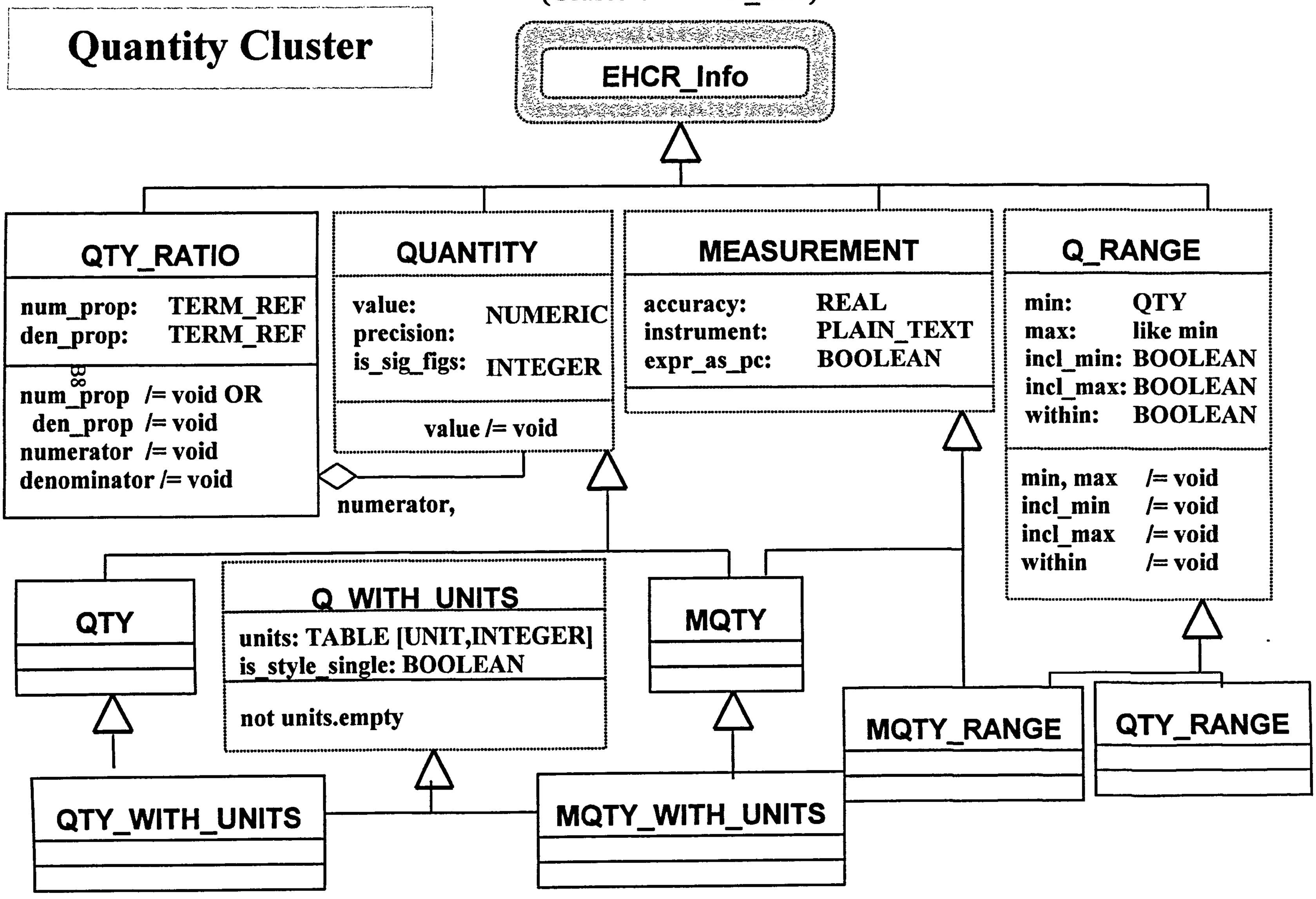


Text Cluster

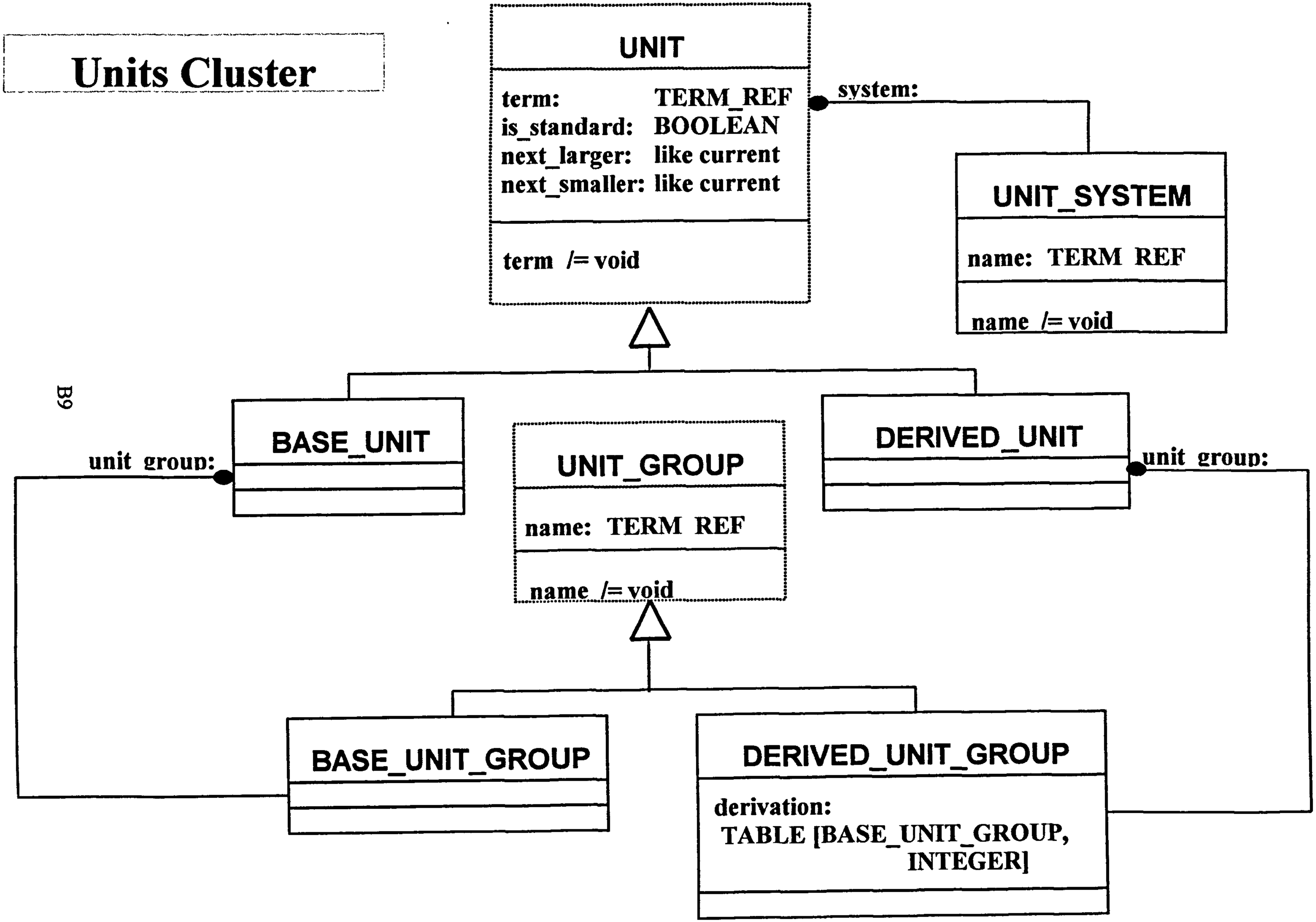


Quantity Cluster

(Cluster: EHCR_Info)

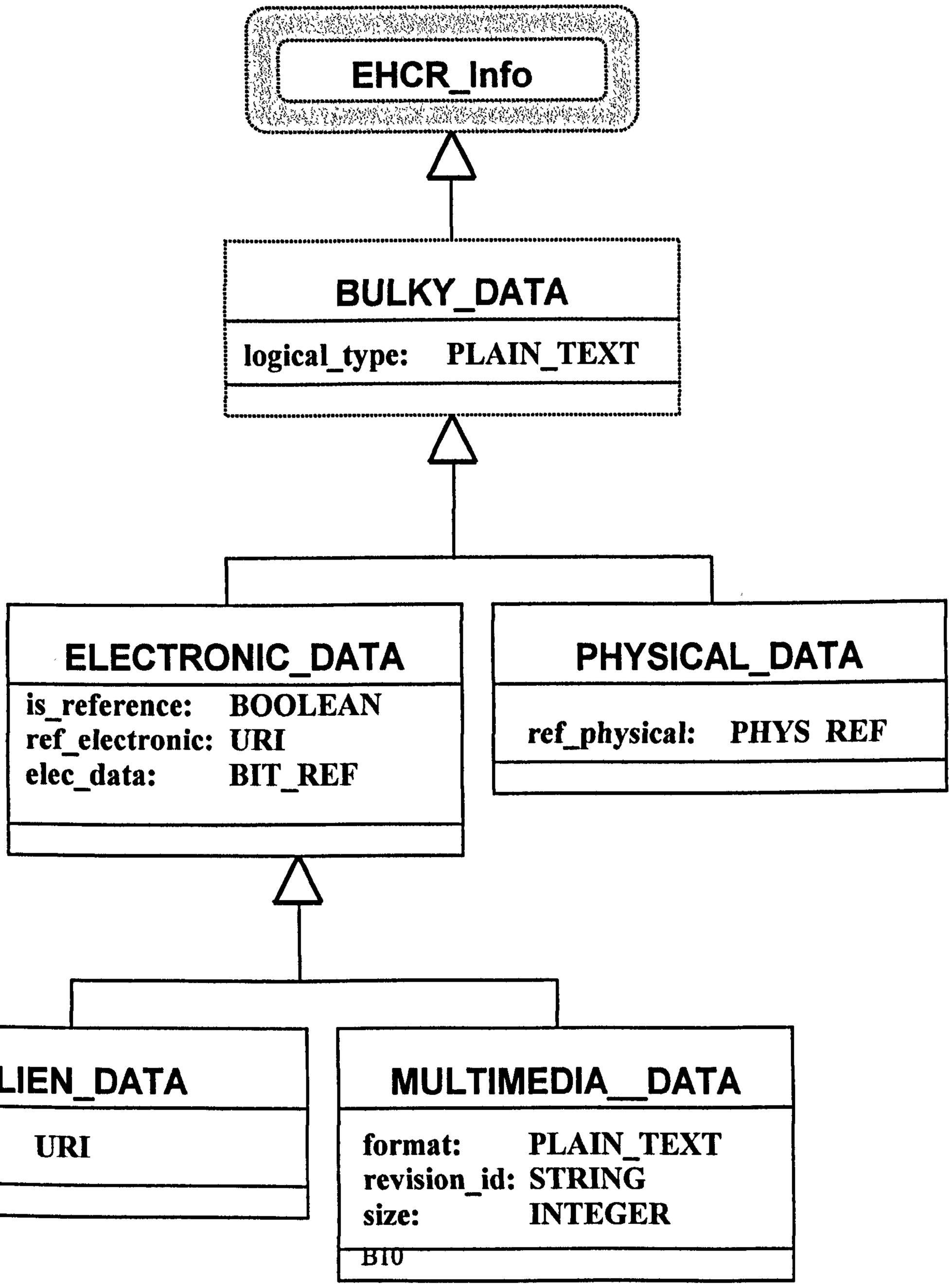


Units Cluster



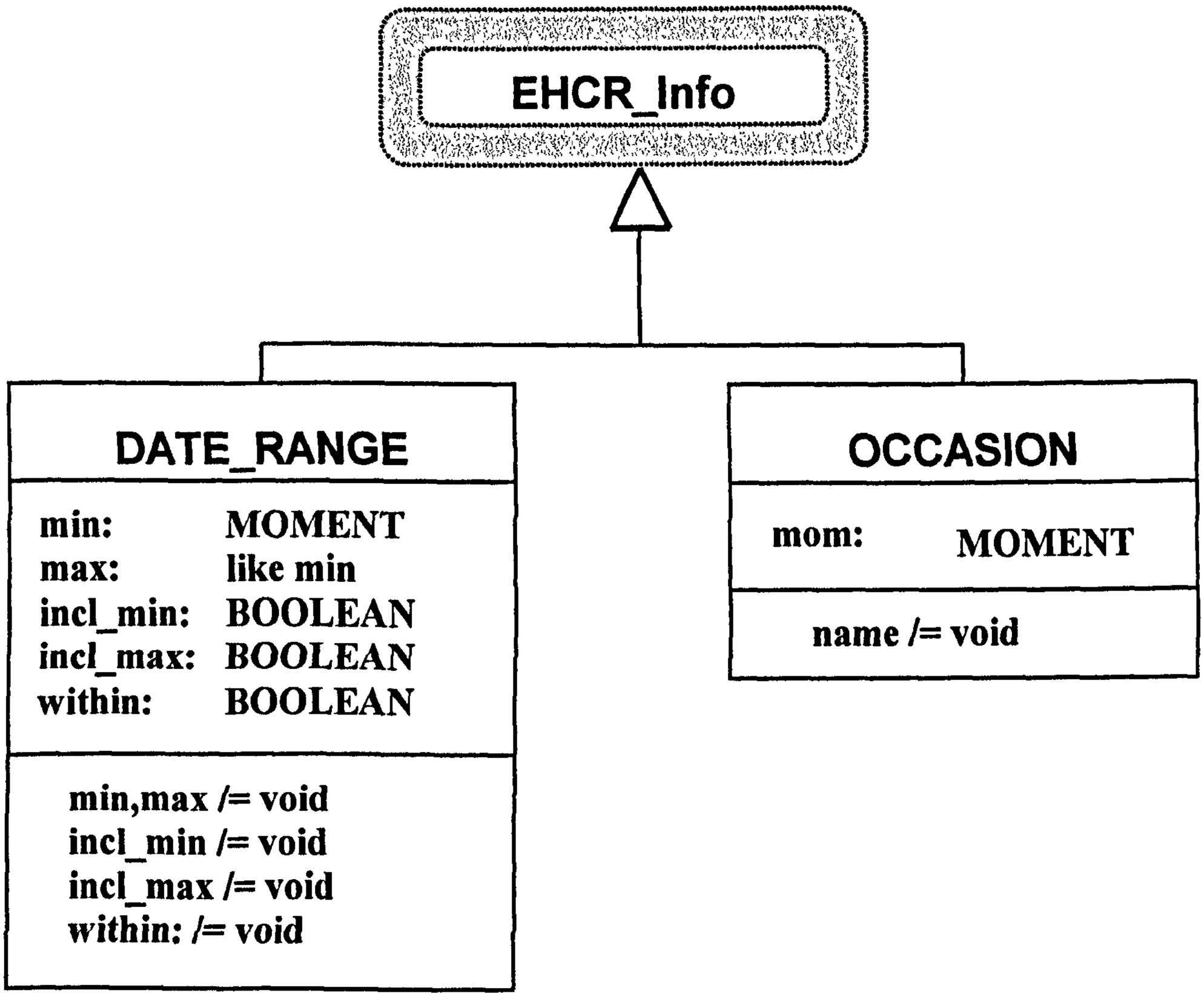
Bulky_Data Cluster

(Cluster: EHCR_Info)

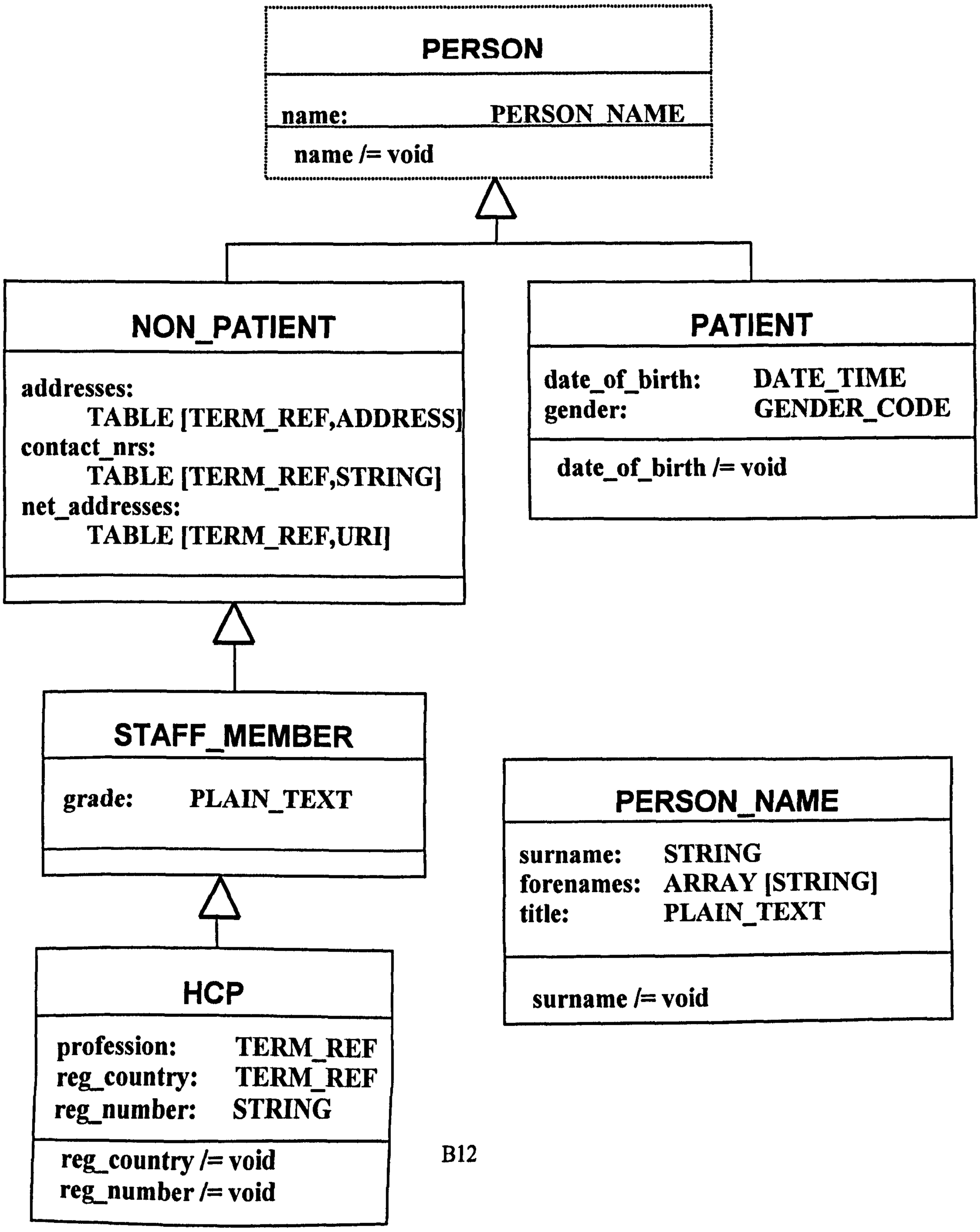


Moment Cluster

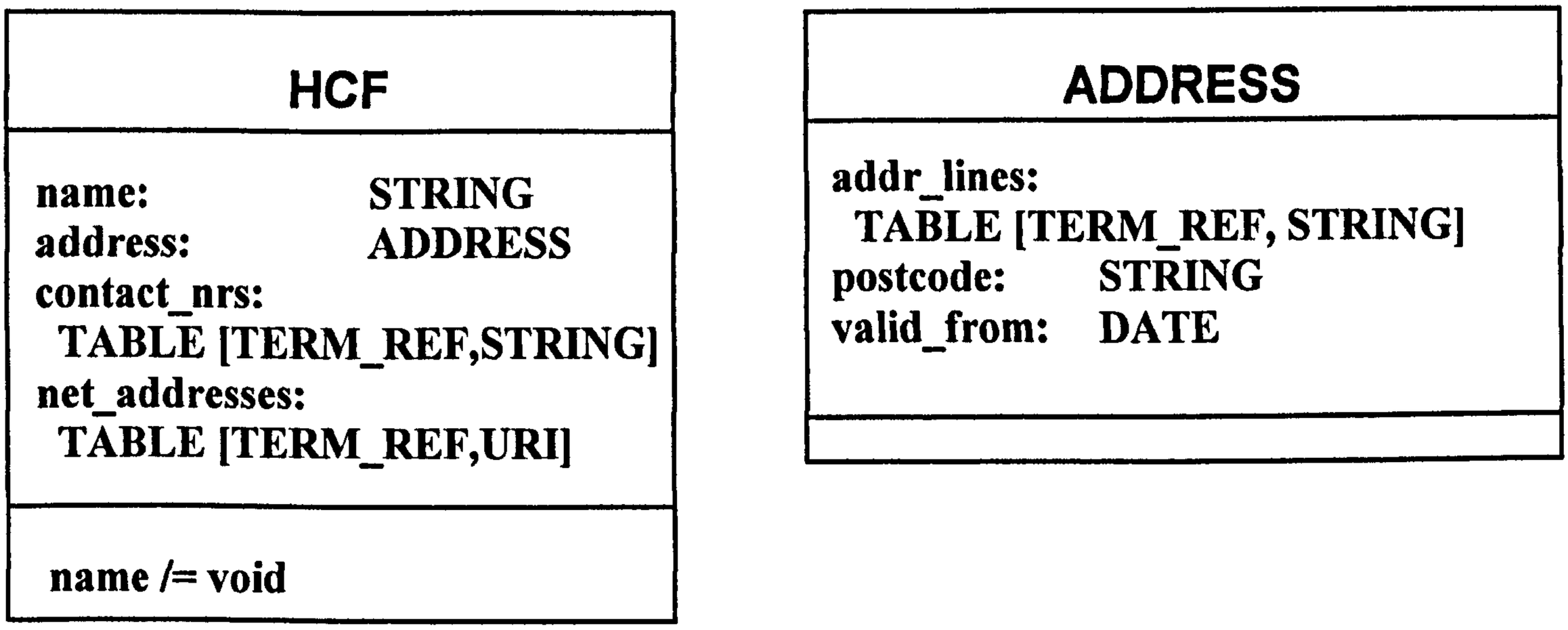
(Cluster: EHCR_Info)



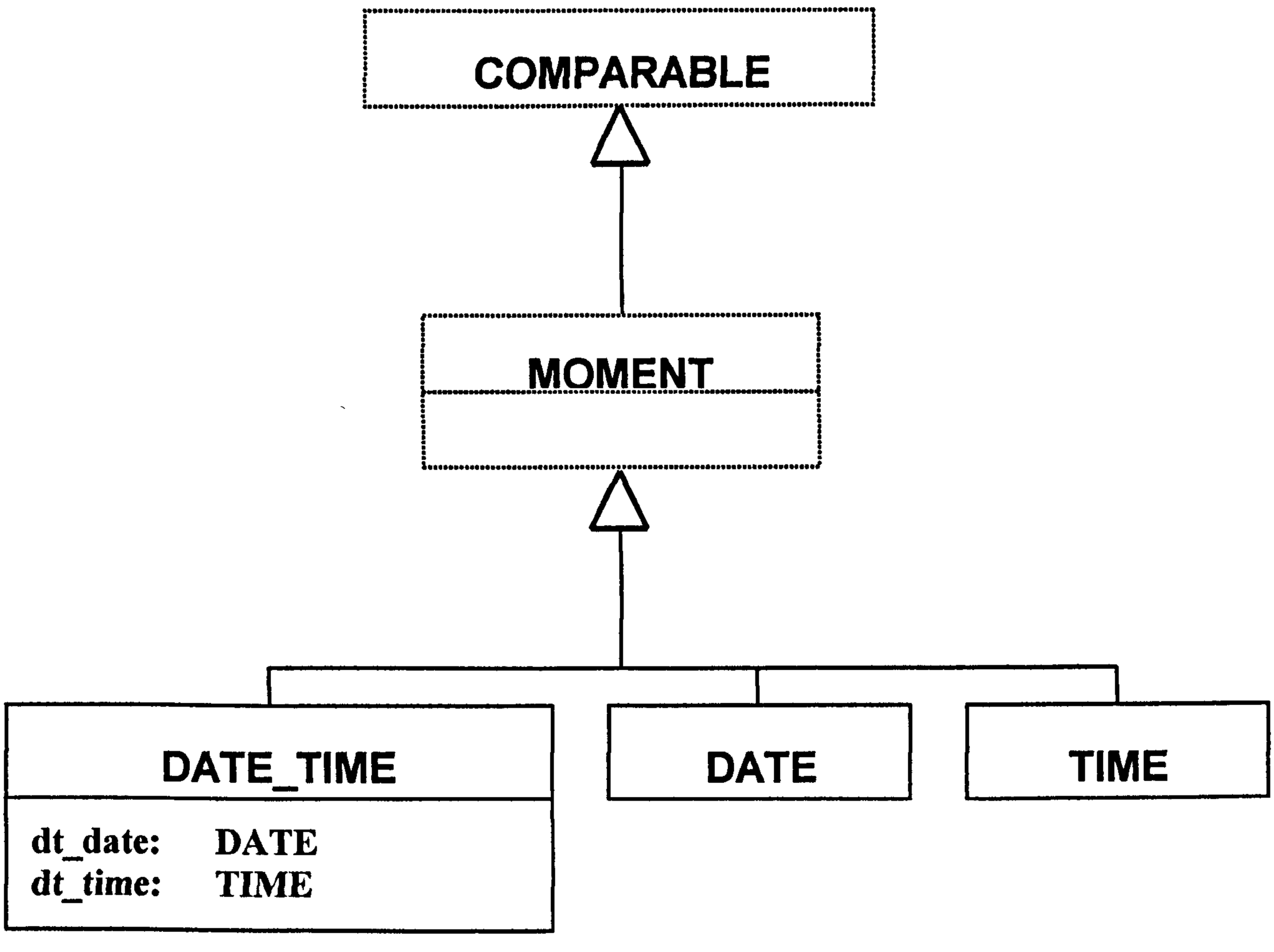
People & Places Cluster

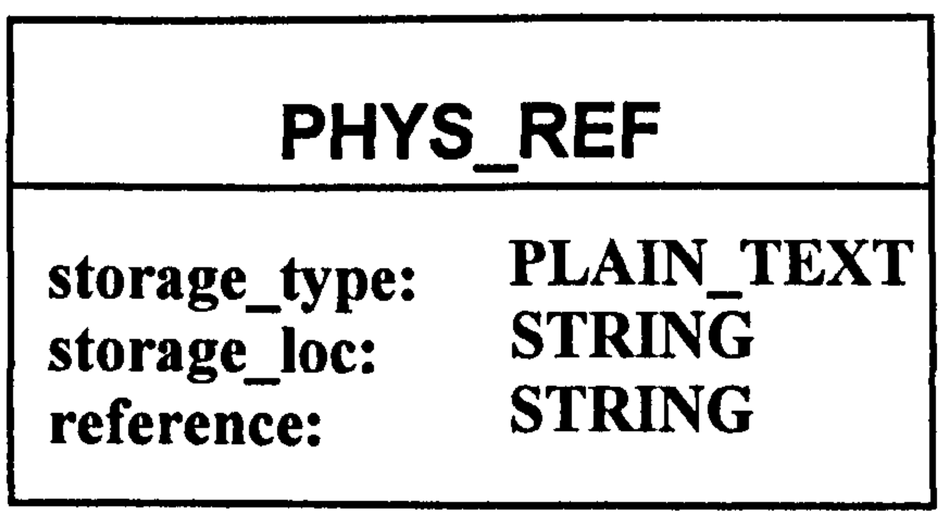


People and Places Cluster (continued)



Basic Cluster





Exchange Cluster

NUMERIC

COMPARABLE

STD_NUMERIC

CHARACTER

BOOLEAN

STRING

INTEGER

REAL

DOUBLE

BIT_REF

CODE_LINK

concept_code: STRING
code_used: STRING

ARRAY [G]

LIST [G]

SET [G]

REVISION

rev: STRING

PERMISSIONS

perm: INTEGER

URI

uri_ref: STRING

GENDER_CODE

gender: INTEGER

GEHR_LANG

lang_code: STRING

EMPH_LEVEL

emph: INTEGER

GEHR_UID

uid: STRING

EHCR_UID

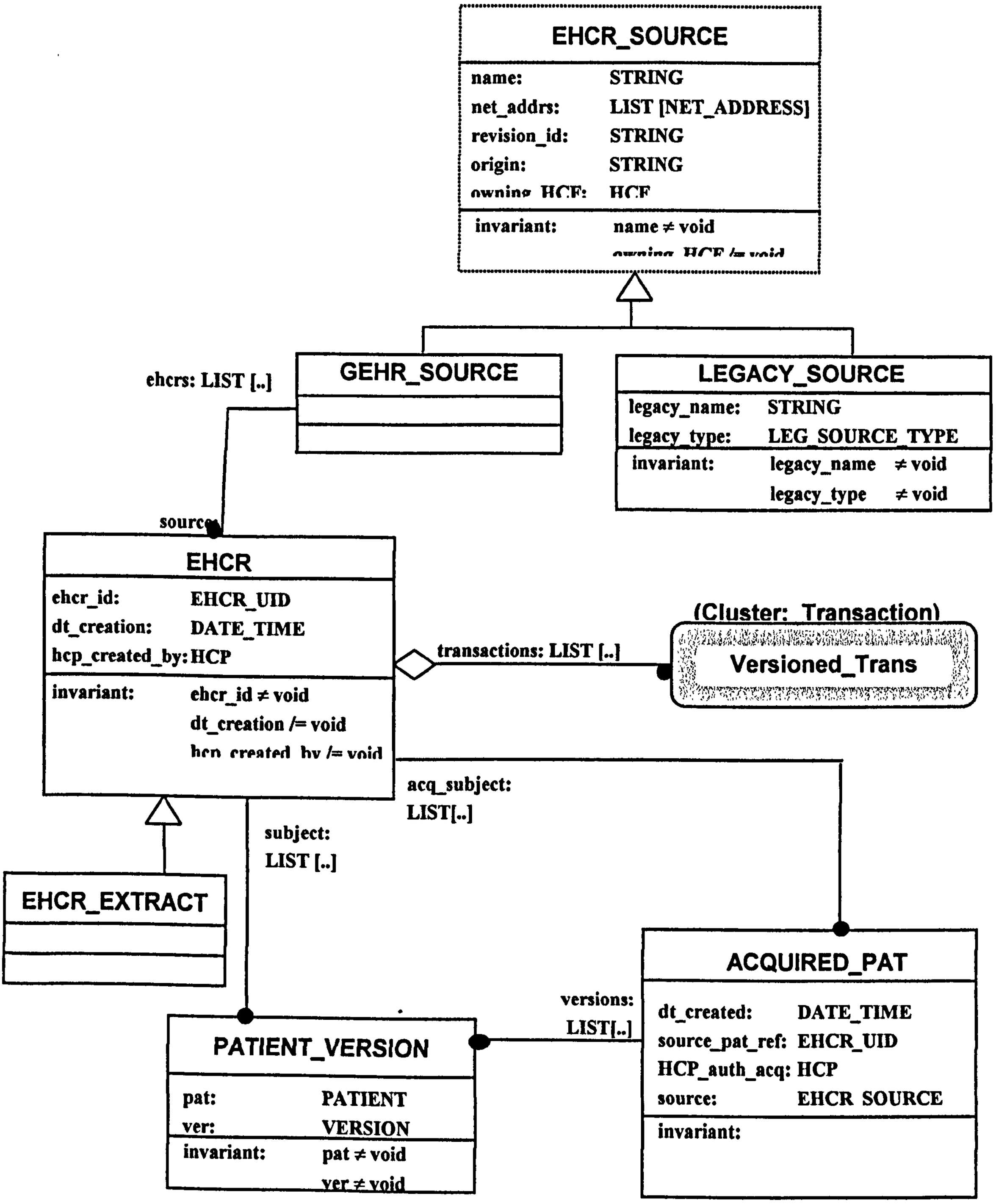
TRANS_UID

ENTRY_UID

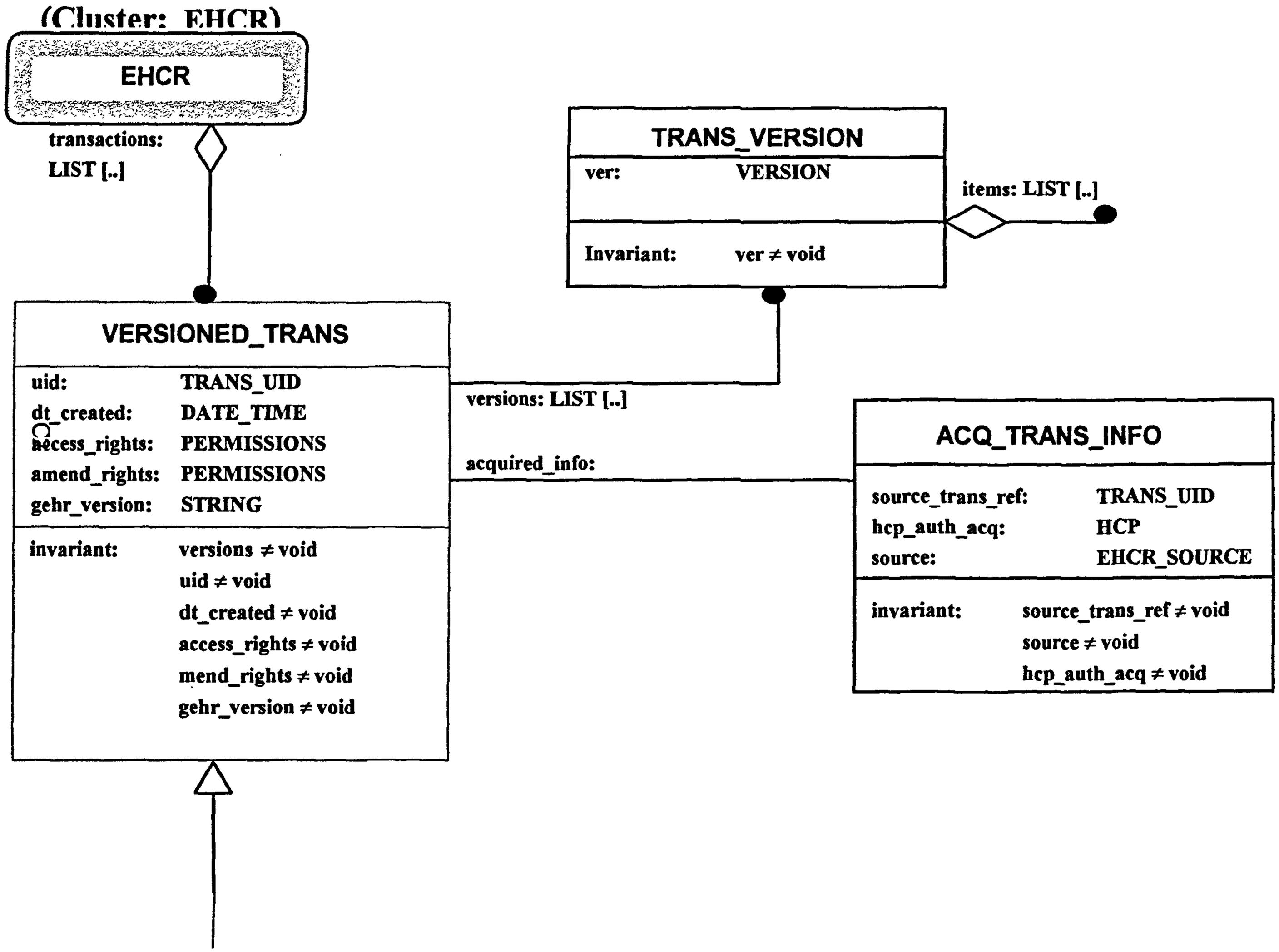
Appendix C – GEHR Object Model v1.5

GEHR Object Module Version 1.5. The following diagrams show the GEHR architecture, see section 4.4.1. The diagrammatic representation of the model is based on the "Rumbaugh" methodology, together with concepts from the Eiffel language and the BON notation.

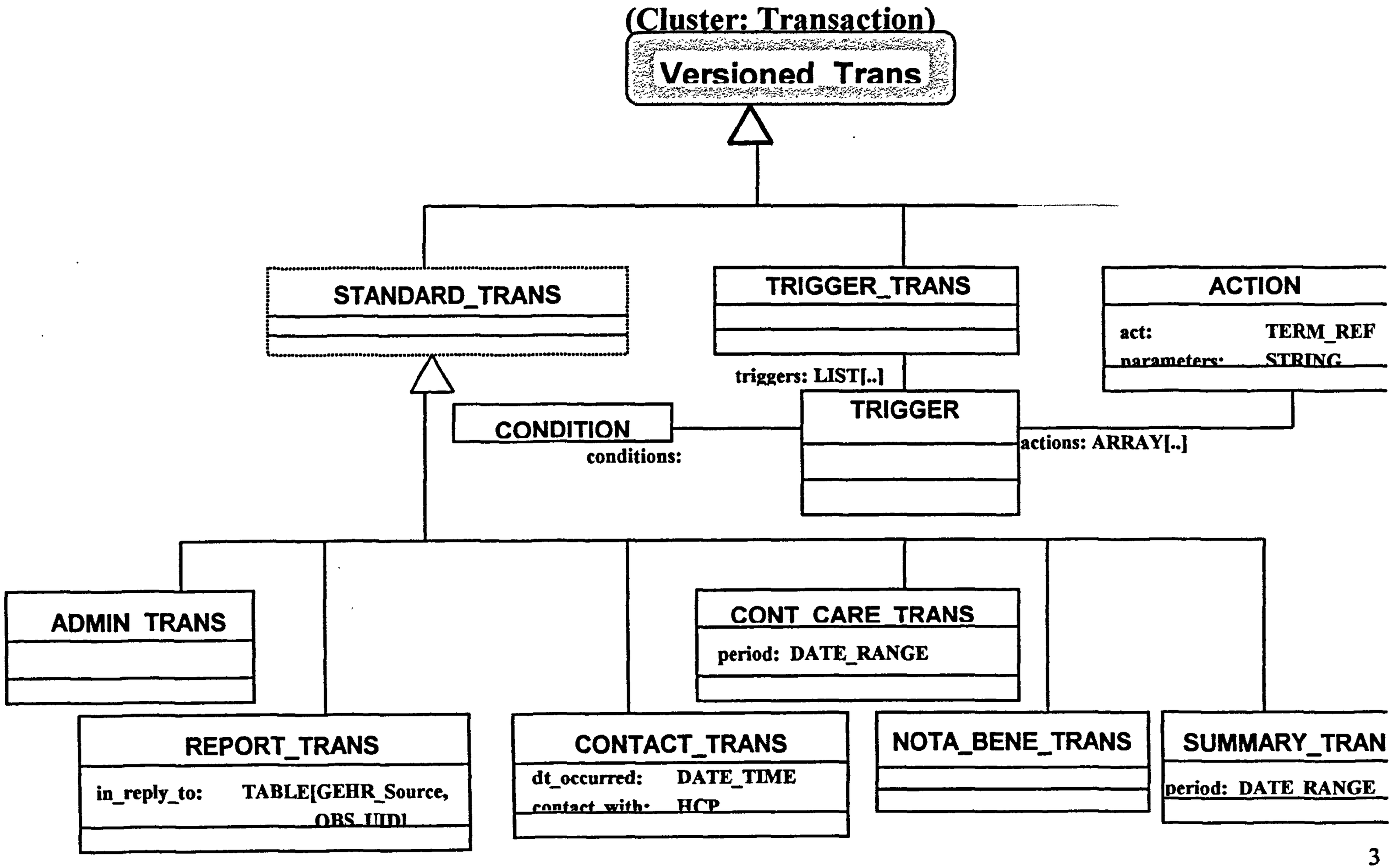
EHCR CLUSTER



TRANSACTION CLUSTER

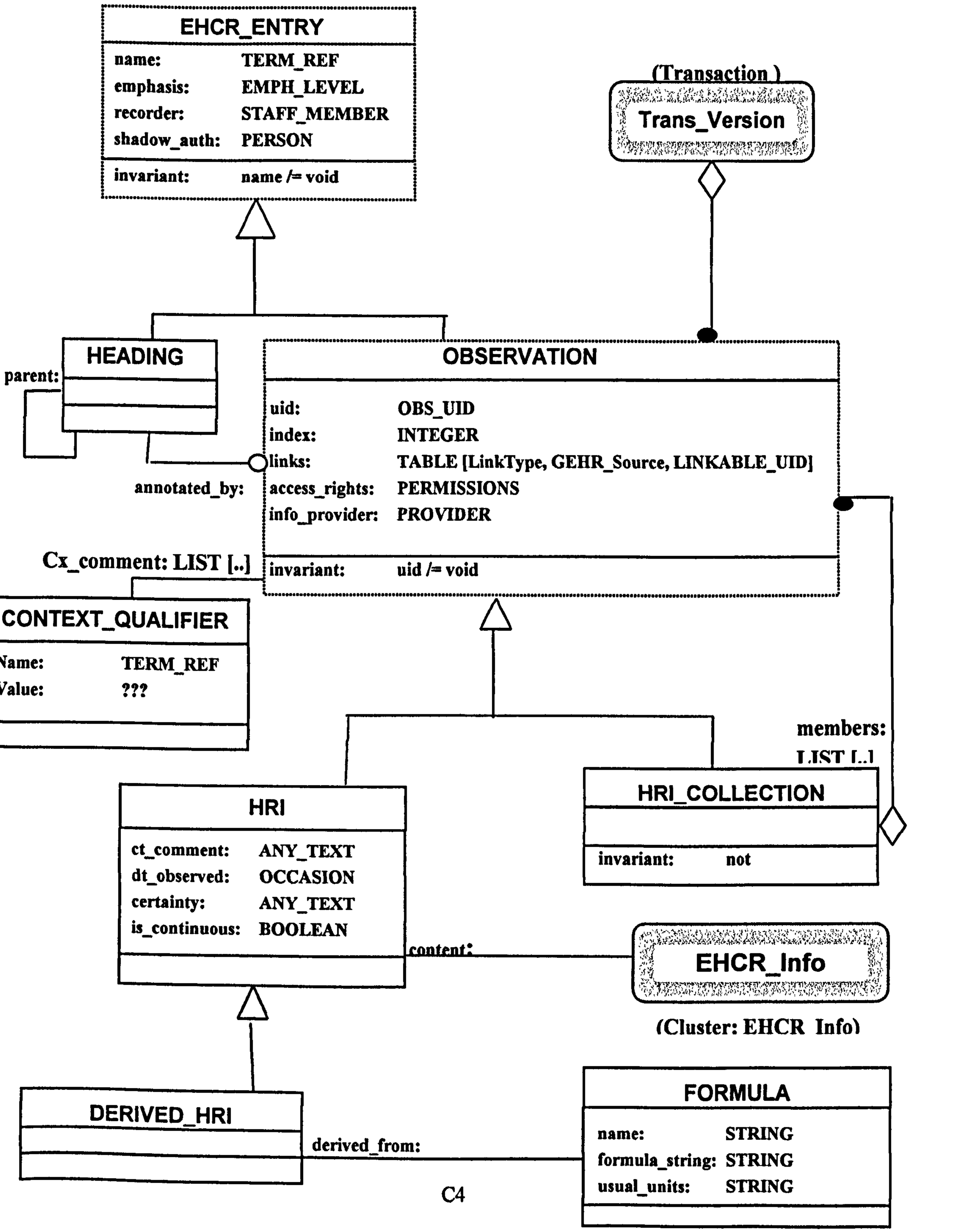


TRANSACTION CLUSTER (CONTINUED)



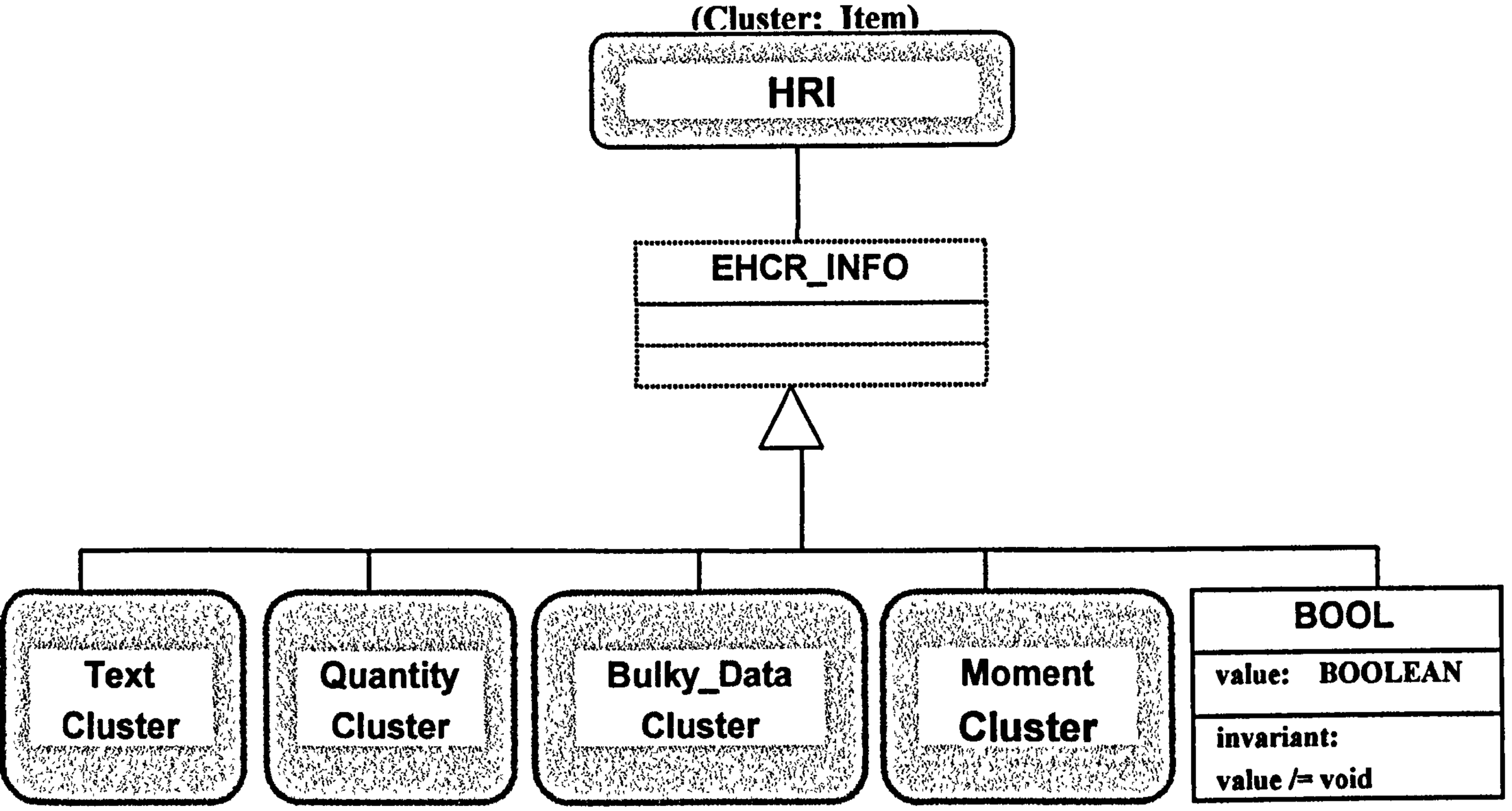
C3

ITEM CLUSTER

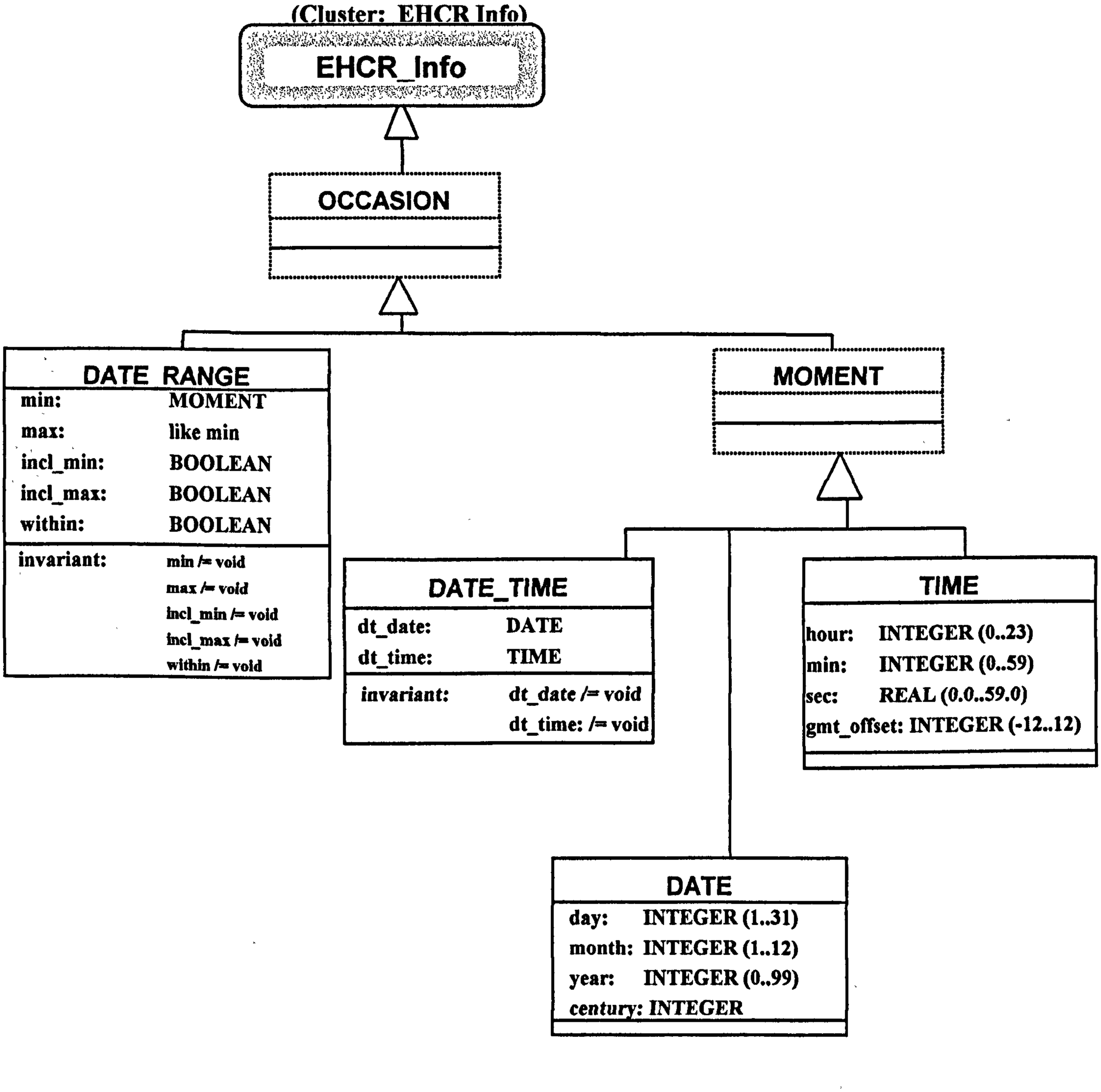


C4

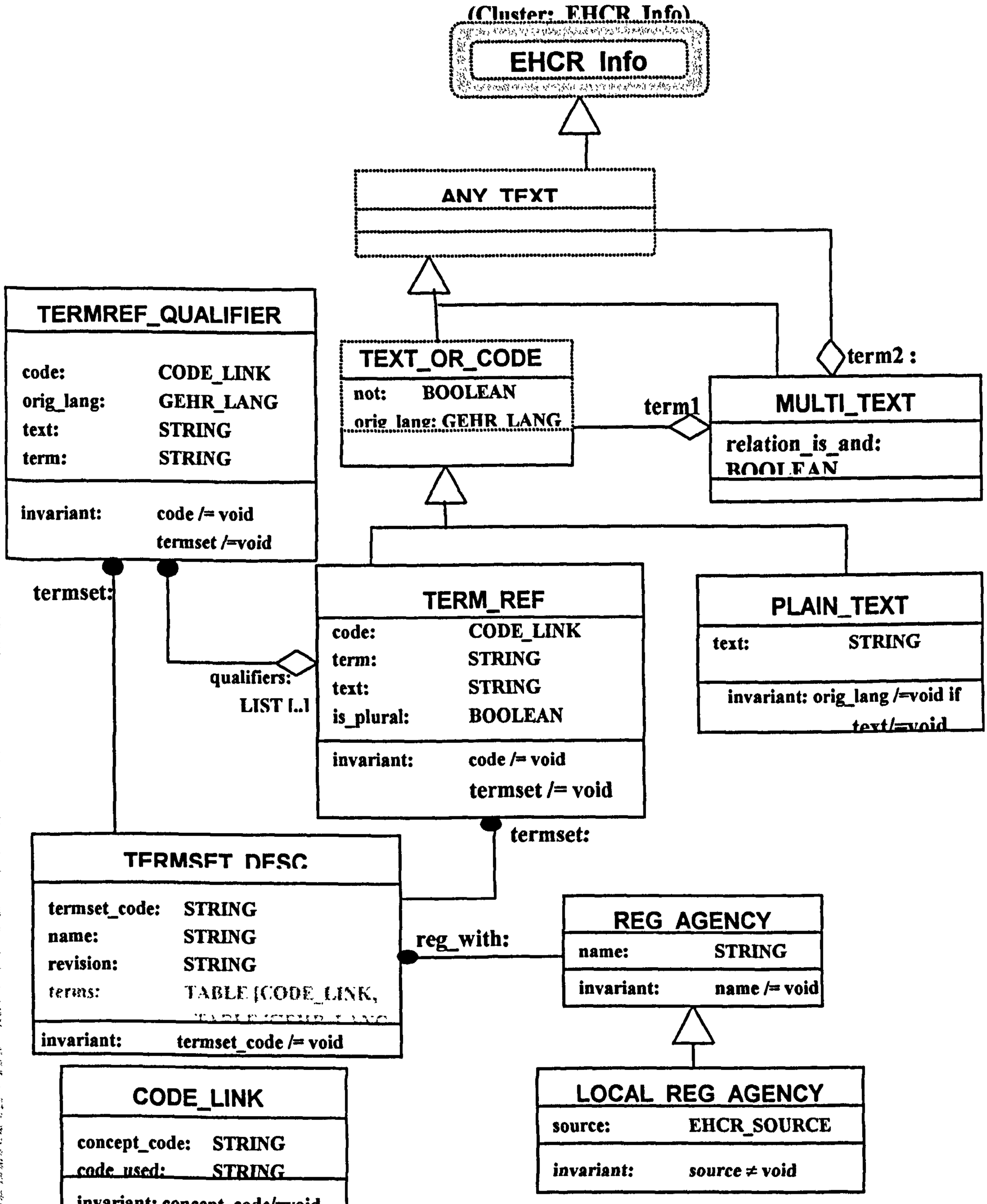
EHCR_INFO CLUSTER



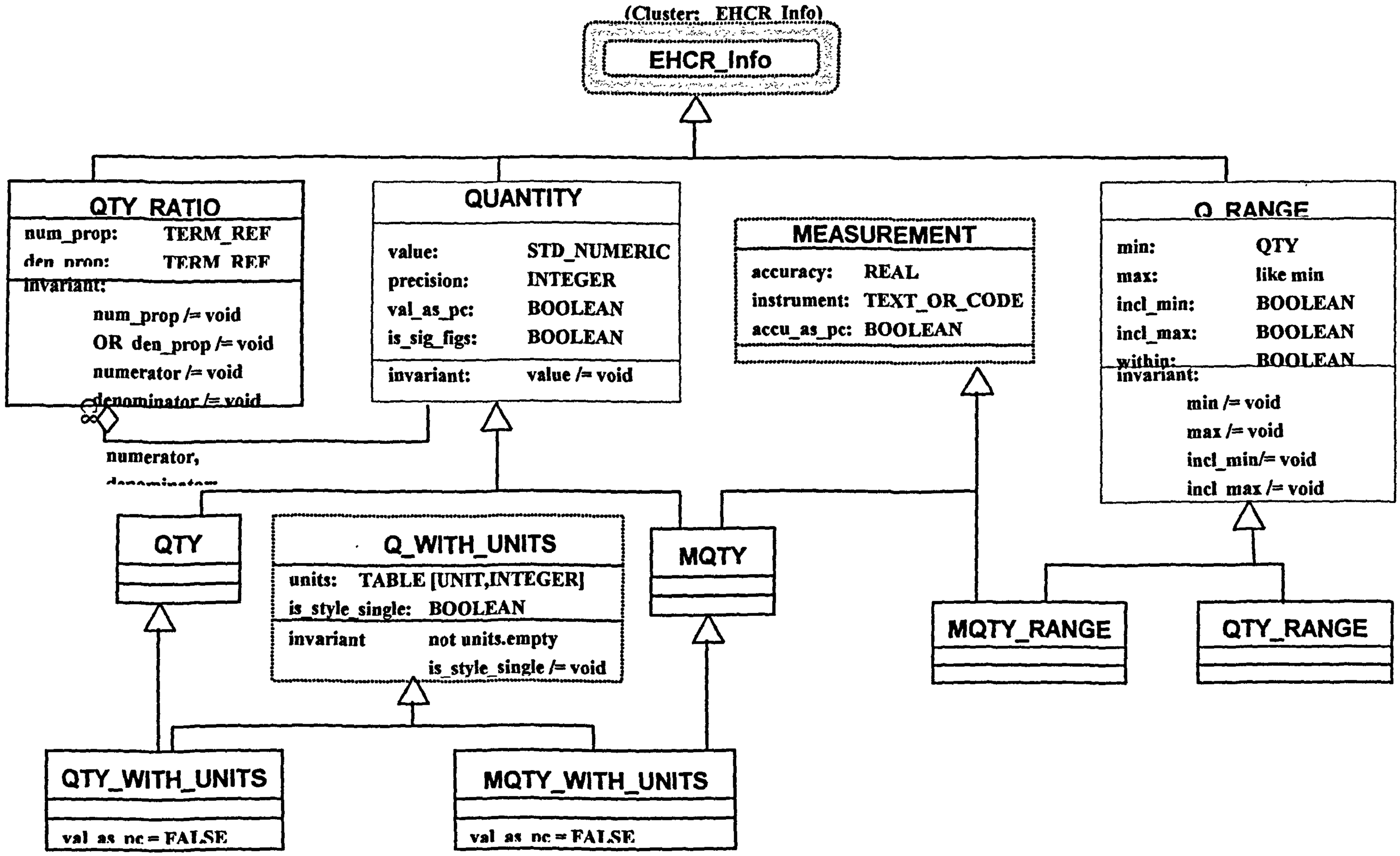
MOMENT CLUSTER



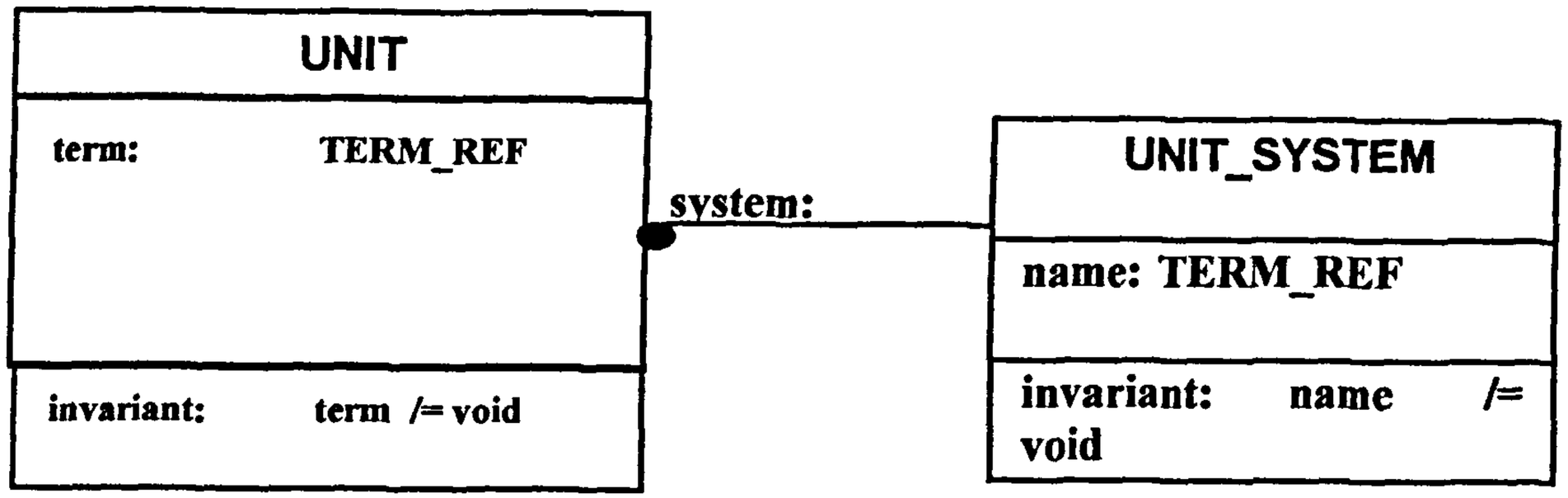
TEXT CLUSTER



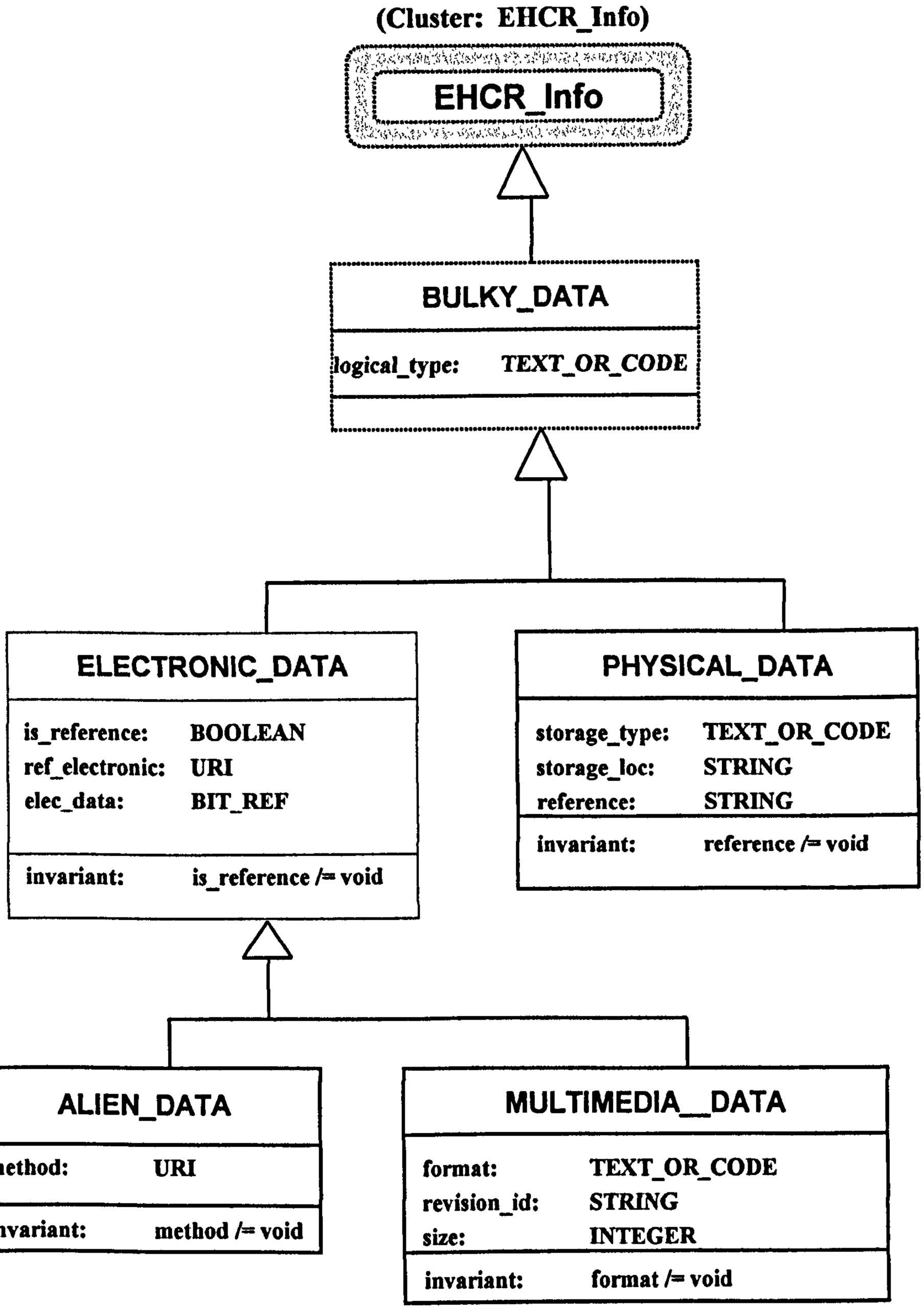
QUANTITY CLUSTER



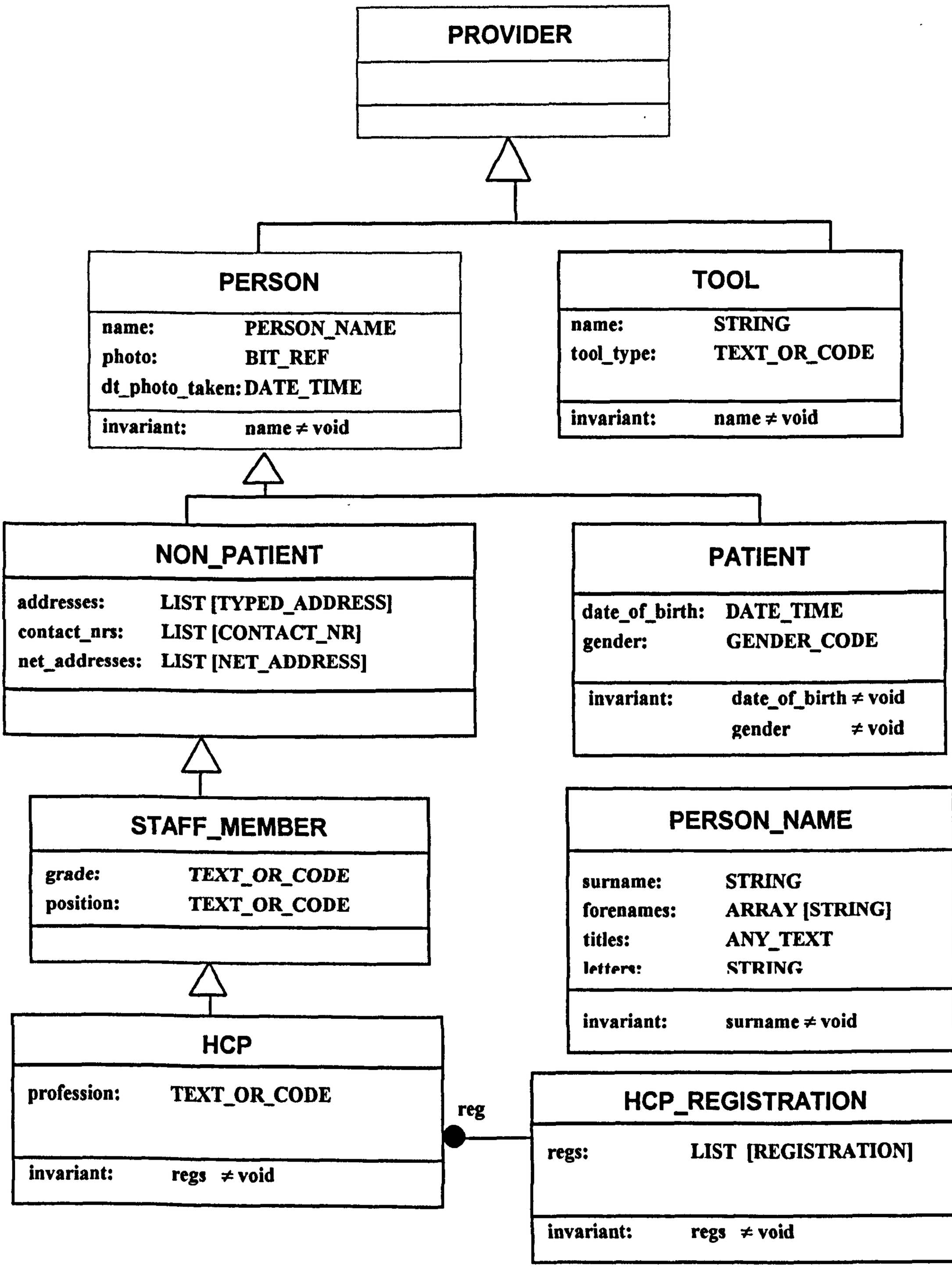
UNITS CLUSTER



BULKY DATA CLUSTER



PEOPLE CLUSTER



PLACES CLUSTER

HCF

name:	STRING
address:	ADDRESS
type_of_hcf:	TERM_REF
reg:	REGISTRATION
contact_nrs:	LIST [CONTACT_NR]
net_addresses:	LIST [NET_ADDRESS]

invariant:	name /= void
	reg /= void

ADDRESS

addr_lines:	LIST [ADDR_LINE]
postcode:	STRING
valid_from:	LIST [DATE]
invalid_from:	LIST [DATE]

ADDRESS_LINE

addr_line_type:	TERM_REF
addr line text:	STRING

CONTACT_NR

contact_nr_type:	TERM_REF
number:	STRING
comment:	STRING
comment_lang:	GEHR_LANG
valid_from:	LIST [DATE]
invalid_from:	LIST [DATE]

invariant:	contact_nr_type ≠ void
	number ≠ void

NET_ADDRESS

net_addr_type:	TERM_REF
net_addr:	URI
comment:	STRING
comment_lang:	GEHR_LANG
valid_from:	LIST [DATE]
invalid_from:	LIST [DATE]

invariant:	net_addr_type ≠ void
	number ≠ void

REGISTRATION

reg_country:	COUNTRY_CODE
reg_type:	TERM_REF
reg_number:	STRING

invariant:	reg_country ≠ void
	reg_type ≠ void
	reg_number ≠ void

TYPED_ADDRESS

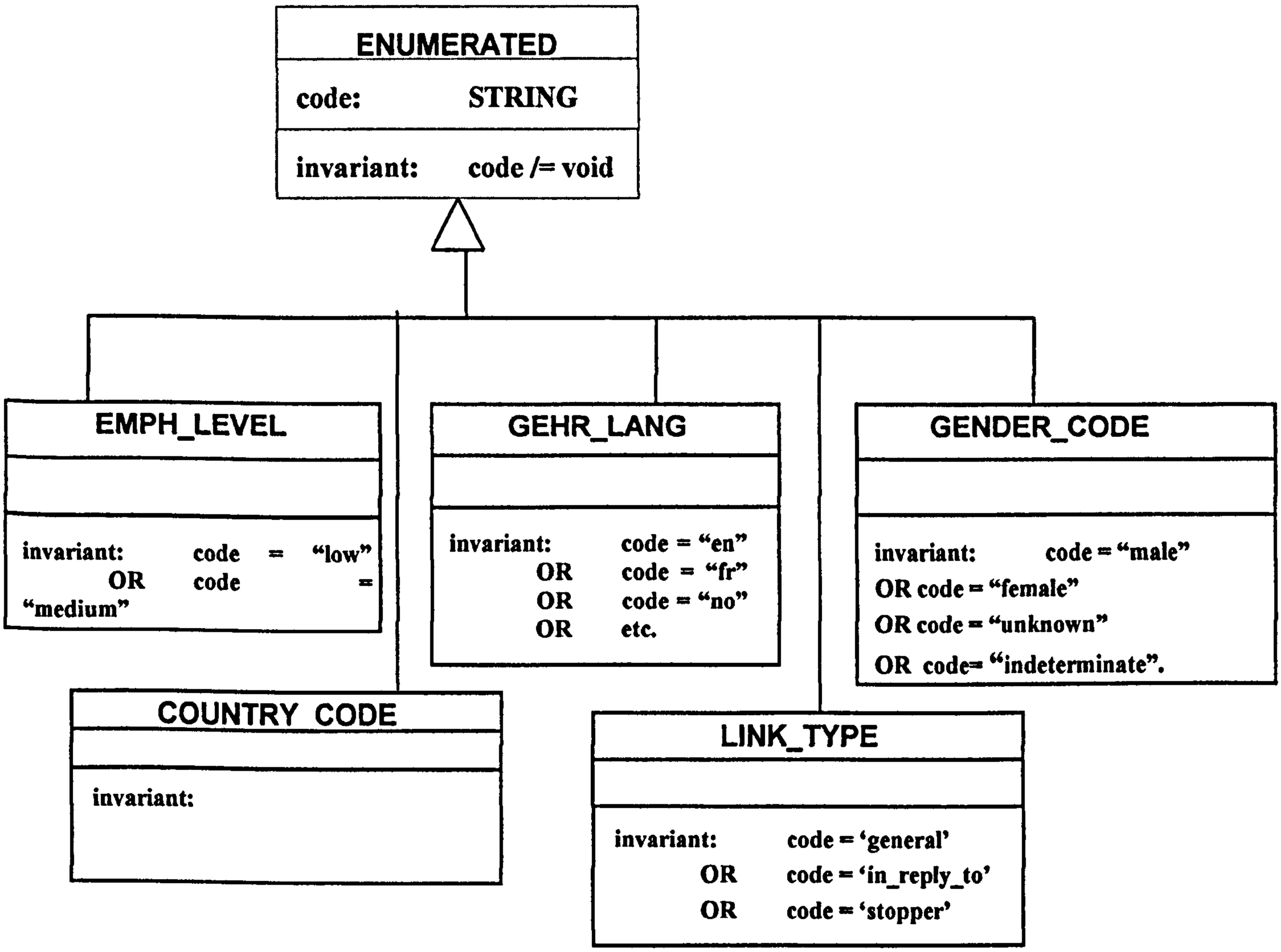
addr_type:	TERM_REF
address:	ADDRESS

invariant:	addr_type ≠ void
	address ≠ void

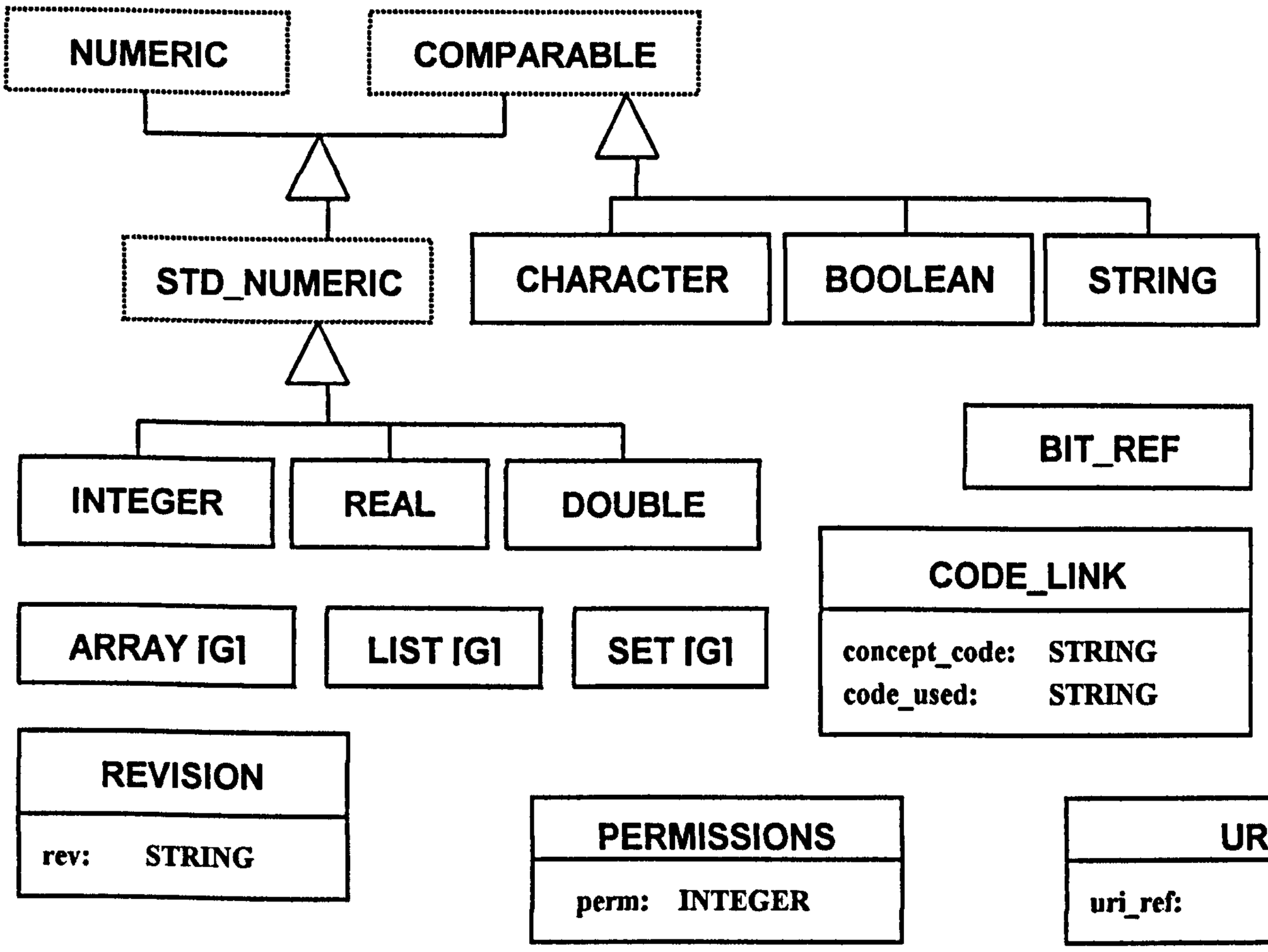
VERSION	
revision:	**REVISION**
dt_committed:	**DATE_TIME**
hcp_authorizing:	**HCP**
hcp_legally_resp:	**HCP**
recorder:	**STAFF MEMBER**
Invariant:	**revision ≠ void**
	dt_committed ≠ void
	hcp_authorizing ≠ void

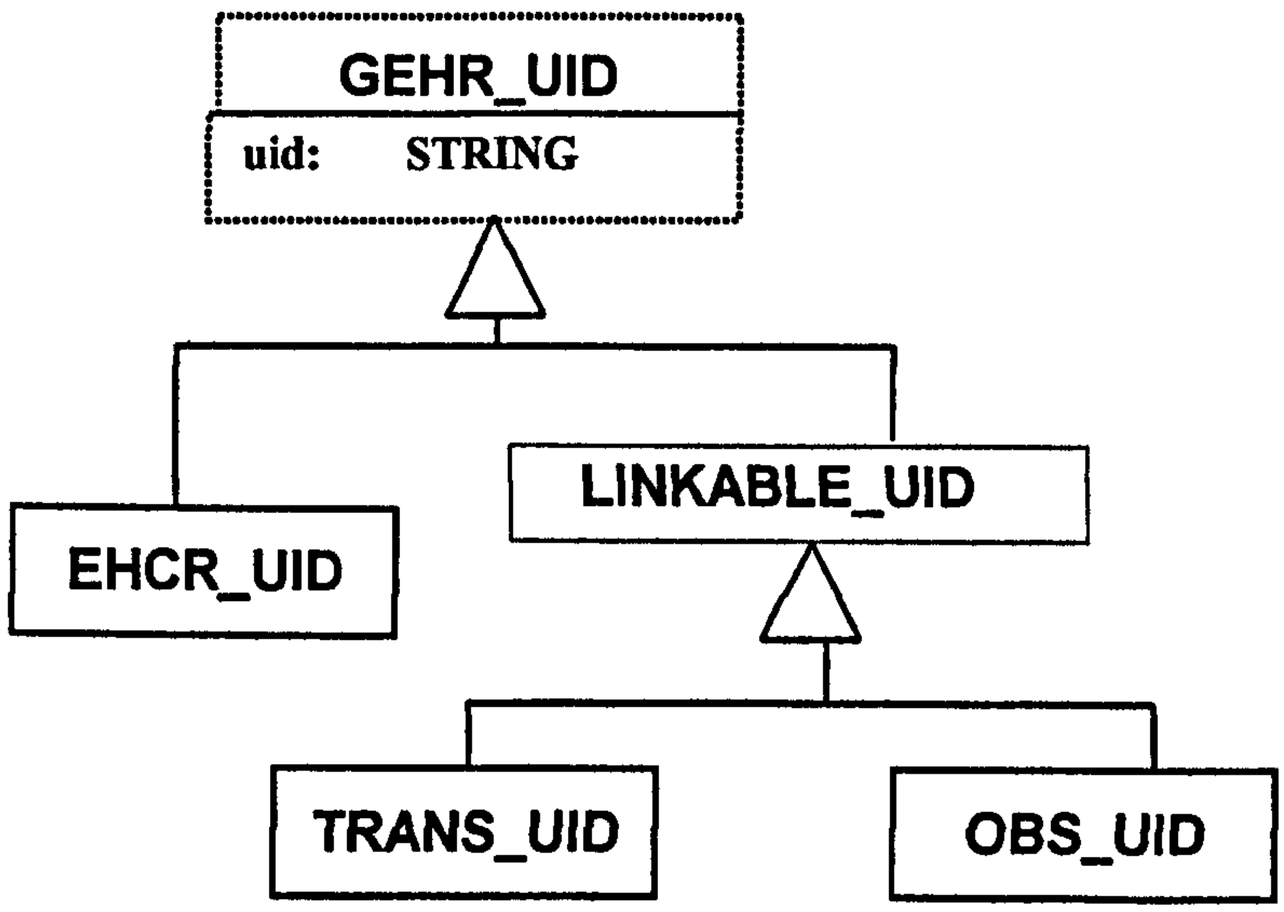
REGISTRATION	
Type_Of_Reg:	**TERM_REF**
Country:	**COUNTRY_CODE**
Reg_Number:	**STRING**
Invariant:	**Country ≠ void**
	Reg_Number ≠ void

ENUMERATED CLUSTER



EXCHANGE CLUSTER





Appendix D – GEHR Exchange Format

The GEHR Exchange Format is presented in ASN.1 format see section 4.6. The GEF is shown in structure sequence.

GEF-2

```
DEFINITIONS ::=
BEGIN
EXPORTS ;
IMPORTS ;

REVISION                ::=     OCTET STRING

PERMISSIONS             ::=     INTEGER

URI                     ::=     OCTET STRING

TRANS-UID               ::=     LINKABLE-UID

OBS-UID                 ::=     LINKABLE-UID

GEHR-UID                ::=     OCTET STRING

EHCR-UID                ::=     GEHR-UID

LINKABLE-UID            ::=     GEHR-UID

DATE                    ::=     OCTET STRING (SIZE (8))

TIME                    ::=     OCTET STRING (SIZE (6))

STD-NUMERIC             ::=     CHOICE {REAL, INTEGER }

BIT-REF                 ::=     BIT STRING

COUNTRY-CODE    ::= OCTET STRING

POINTER-TO-ITEMS        ::=     CHOICE {
                [0]     OBSERVATIONID,
                [1]     HRICOLLECTIONID,
                [2]     HRIID }

POINTER-TO-STAFF-MEMBER ::=     CHOICE {
                [0]     STAFFMEMBERID,
                [1]     HCPID }

POINTER-TO-HCP  ::=     POINTER

POINTER-TO-HCF  ::=     POINTER

POINTER-TO-EHCR-SOURCE      ::=     POINTER
```

```
POINTER-TO-PERSON       ::=     CHOICE{
                [0]     PERSONID,
                [1]     NONPATIENTID,
                [2]     PATIENTID }

POINTER-TO-PATIENT      ::=     POINTER

POINTER-TO-HEADING      ::=     POINTER

POINTER-TO-REG-AGENCY   ::=     POINTER

POINTER-TO-TERMSET      ::=     POINTER

EHCRID                  ::=     ID

STAFFMEMBERID           ::=     ID

HCPID                   ::=     ID

HCFID                   ::=     ID

PERSONID                ::=     ID

NONPATIENTID            ::=     ID

PATIENTID               ::=     ID

REG-AGENCYID            ::=     ID

POINTER                 ::=     INTEGER

ID                      ::=     INTEGER

EHCR-SOURCE     ::= [PRIVATE 40] SET {
        ehcrsourceid            [0]     ID,
        name                    [1]     OCTET STRING,
        net-addrs               [2]     SET OF NET-ADDRESS      OPTIONAL,
        revision-id             [3]     OCTET STRING            OPTIONAL,
        origin                  [4]     OCTET STRING            OPTIONAL,
        owning-hcf              [5]     POINTER-TO-HCF  }

GEHR-SOURCE     ::= [PRIVATE 42] SET
        {               EHCR-SOURCE}

LEGACY-SOURCE   ::= [PRIVATE 43] SET
```

```
        {                       EHCR-SOURCE,
        legacy-name             [0]     OCTET STRING,
        legacy-type             [1]     LEG-SOURCE-TYPE }

EHCR ::= [PRIVATE 41] SET {
        ehcr-id                 [0]     EHCR-UID,
        dt-creation             [1]     DATE-TIME,
        hcp-created-by          [2]     POINTER-TO-HCP,
        transactions            [3]     SET OF CHOICE {ADMIN, ADMIN-
        SUMMARY, CONTACT,  CONT-CARE, NOTABENE, SUMMARY,
        REPORT} }
        acq-subject             [4]     SET OF ACQUIRED-PAT    OPTIONAL,
        subject                 [5]     SET OF PATIENT-VERSION
        OPTIONAL,
        source                  [6]     CHOICE {GEHR-SOURCE, LEGACY-
        SOURCE}     }

EHCR-EXTRACT    ::= [PRIVATE 44] SET {
                    EHCR }

PATIENT-VERSION ::= [PRIVATE 45] SET {
        pat                     [0]     PATIENT,
        ver                     [1]     VERSION }

ACQUIRED-PAT    ::= [PRIVATE 46] SET {
        dt-created              [0]     DATE-TIME               OPTIONAL,
        source-pat-ref          [1]     EHCR-UID                OPTIONAL,
        hcp-auth-acq            [2]     POINTER-TO-HCP          OPTIONAL,
        source                  [3]     POINTER-TO-EHCR-SOURCE
        OPTIONAL,
        versions                [4]     SET OF PATIENT-VERSION
        OPTIONAL }

VERSIONED-TRANS         ::= [PRIVATE 0] SET {
        uid                     [0]     TRANS-UID,
        dt-created              [1]     DATE-TIME,
        access-rights           [2]     PERMISSIONS,
        amend-rights            [3]     PERMISSIONS,
        gehr-version            [4]     OCTET STRING,
        acquired-info           [5]     ACQ-TRANS-INFO OPTIONAL,
        versions                [6]     SET OF TRANS-VERSION }

TRANS-VERSION   ::= [PRIVATE 1] SET {
        ver                     [0]     VERSION,
        items                   [1]     SET OF CHOICE {HRI, HRI-COLLECTION,
HEADING, DERIVED-HRI} OPTIONAL }
```

```
ACQ-TRANS-INFO  ::= [PRIVATE 47] SET {
        source-trans-ref        [0]     TRANS-UID,
        hcp-auth-acq            [1]     POINTER-TO-HCP,
        source                  [2]     POINTER-TO-EHCR-SOURCE }


STANDARD-TRANS          ::= [PRIVATE 2] SET {
                        VERSIONED-TRANS }

REPORT      ::= [PRIVATE 3] SET
                {                               STANDARD-TRANS,
                 in-reply-to    [0]     SEQUENCE {POINTER-TO-EHCR-SOURCE,
                                                  OBS-UID }    OPTIONAL
                }

CONT-CARE ::= [PRIVATE 4] SET
        {                                       STANDARD-TRANS,
        period                  [0]     DATE-RANGE      OPTIONAL }

TRIGGER     ::= [PRIVATE 5]     SET
        {                       VERSIONED-TRANS,
         triggers               [0]     SET OF GO OPTIONAL}

GO      ::= [PRIVATE 6]     SET
        {actions        SET OF ACTION OPTIONAL}

ACTION      ::= [PRIVATE 7]     SET
        {act                    [0]     TERM-REF        OPTIONAL,
         parameters             [1]     OCTET STRING    OPTIONAL    }

CONDITION  ::= [PRIVATE 8]      SET
        {conditions             GO OPTIONAL}

SUMMARY    ::= [PRIVATE 9]      SEQUENCE
        {                               STANDARD-TRANS,
        period                  [0]     DATE-RANGE      OPTIONAL}

ADMIN       ::= [PRIVATE 10]    SEQUENCE
        {                               STANDARD-TRANS}

ADMIN-SUMMARY ::= [PRIVATE 11]      SEQUENCE
        {                               STANDARD-TRANS,
         period                 [0] DATE-RANGE    OPTIONAL }
```

```
NOTABENE  ::= [PRIVATE 12]     SEQUENCE
        {                                STANDARD-TRANS }

CONTACT    ::= [PRIVATE 13]     SET
        {                                STANDARD-TRANS,
        dt-occurred              [0]     DATE-TIME          OPTIONAL,
        contact-with             [1]     POINTER-TO-HCP     OPTIONAL }

EHCR-ENTRY        ::= [PRIVATE 14]     SET
        {id                              [0]     EHCR-ENTRY-ID,
        uid                      [1]     EHCR-UID,
        name                     [2]     TERM-REF,
        emphasis                 [3]     EMPH-LEVEL
        OPTIONAL,
        recorder                 [4]     POINTER-TO-STAFF-MEMBER
        OPTIONAL,
        shadowauth               [5]     POINTER-TO-PERSON
        OPTIONAL}

HEADING    ::= [PRIVATE 15]     SET
        {                                EHCR-ENTRY,
        parent  [0]     POINTER-TO-HEADING     OPTIONAL }

OBSERVATION       ::= [PRIVATE 16]     SET
        {                                EHCR-ENTRY,
         uid                     [0]     OBS-UID,
        links                    [1]     SET OF LINKABLE-UID OPTIONAL,
        accessrights             [2]     PERMISSIONS        OPTIONAL,
        cxcomment                [3]     MULTI-TEXT         OPTIONAL,
        inreplyto                [4]     SEQUENCE{GEHR-SOURCE, LINKABLE-
UID}OPTIONAL,
        info-provider            [5]     PROVIDER           OPTIONAL,
        annotatedby              [6]     SET OF HEADING, POINTER-TO-HEADING
        OPTIONAL }

HRI-COLLECTION  ::=[PRIVATE 17]     SET
        {                                OBSERVATION,
        members                  [0]     SET OF CHOICE {HEADING, HRI-
COLLECTION, HRI}}

HRI    ::= [PRIVATE 18]     SET
        {                                OBSERVATION,
        ctcomment                [0]     MULTI-TEXT         OPTIONAL,
        dtobserved               [1]     DATE-TIME  OPTIONAL,
        certainty                [2]     PLAIN-TEXT OPTIONAL,
```

```
        ctemphasis              [3]     EMPH-LEVEL          OPTIONAL,
        content                 [4]  CHOICE {BOOLEAN, OCCASSION, DATE-
RANGE, MULTI-TEXT, PLAIN-TEXT, TERM-REF, QUANTITY, QTY-RATIO,
QTY, QTY-WITH-UNITS,MQTY, MQTY-WITH-UNITS, MQTY-RANGE, QTY-
RANGE, PHYSICAL-DATA, ALIEN-DATA, MULTIMEDIA-DATA }    }


DERIVED-HRI         ::= [PRIVATE 19]     SET
        {                                HRI,
        derived-from            [0]      FORMULA OPTIONAL}

FORMULA     ::= [PRIVATE 20]     SET
        {name                   [0]      OCTET STRING OPTIONAL,
        formula-string          [1]      OCTET STRING OPTIONAL,
        usual-units             [2]      OCTET STRING OPTIONAL}

EHCR-INFO   ::= [PRIVATE 21]     SET
        {       }

PLAIN-TEXT ::=[PRIVATE 22]       SET
        {                                EHCR-INFO,
        value                   [0]      OCTET STRING                OPTIONAL,
        orig-lang               [1]      GEHR-LANG OPTIONAL}

MULTI-TEXT          ::= [PRIVATE 23]     SEQUENCE
        {                        EHCR-INFO,
        value                   [0]      SEQUENCE OF PLAIN-TEXT OPTIONAL}

TERM-REF    ::= [PRIVATE 24]     SET
        {                                PLAIN-TEXT,
        code                    [0]      CODE-LINK,
        is-plural               [1]      BOOLEAN    OPTIONAL,
        qualifiers              [2]      SET OF TERMREF-QUALIFIER OPTIONAL,
        termset                 [3]      POINTER-TO-TERMSET OPTIONAL }

TERMSET     ::= [PRIVATE 25]     SET
        {termset-code           [0]      OCTET STRING,
        name                    [1]      OCTET STRING      OPTIONAL,
        revision                [2]      OCTET STRING      OPTIONAL,
        terms-used              [3]      SEQUENCE {CODE-LINK, SEQUENCE
{GEHR-LANG,  OCTET STRING } } OPTIONAL,
reg-with                        [4]      CHOICE {REG-AGENCY, LOCAL-REG-
AGENCY}     OPTIONAL }

REG-AGENCY          ::= [PRIVATE 26]     SET
        {regagencyid            [0]      REG-AGENCYID,
```

```
        name                    [1]     OCTET STRING }

TERMREF-QUALIFIER       ::= [PRIVATE 27]     SET
        {code                   [0]     CODE-LINK,
        orig-lang               [1]     GEHR-LANG OPTIONAL,
        termset                 [2]     POINTER-TO-TERMSET
        OPTIONAL }

LOCAL-REG-AGENCY        ::= [PRIVATE 28]     SET
        {                       REG-AGENCY,
        source                  [0]     CHOICE {GEHR-SOURCE, LEGACY-
SOURCE}

QTY-RATIO  ::= [PRIVATE 29]     SET
        {                       EHCR-INFO,
        num-prop                [0]     TERM-REF,
        den-prop                [1]     TERM-REF,
        numerator               [2]     QUANTITY,
        denominator             [3]     QUANTITY }

QUANTITY   ::= [PRIVATE 30]     SET
        {                       EHCR-INFO,
        value                   [0]     STD-NUMERIC,
        precision               [1]     INTEGER         OPTIONAL,
        val-as-pc               [2]     BOOLEAN    OPTIONAL,
        is-sig-figs             [3]     BOOLEAN    OPTIONAL }

Q-RANGE     ::= [PRIVATE 31]    SET
        {                               EHCR-INFO,
        min                     [0]     QTY,
        max                     [1]     QTY,
        incl-min                [2]     BOOLEAN,
        incl-max                [3]     BOOLEAN,
        within                  [4]     BOOLEAN}

MEASUREMENT     ::= [PRIVATE 32]     SET
        { accuracy              [0]     STD-NUMERIC     OPTIONAL,
        instrument              [1]     PLAIN-TEXT OPTIONAL,
        accu-as-pc              [2]     BOOLEAN    OPTIONAL }

QTY    ::= [PRIVATE 33]     SET
        {QUANTITY }

MQTY::= [PRIVATE 34]     SEQUENCE
        {QUANTITY, MEASUREMENT }
```

```
Q-WITH-UNITS        ::= [PRIVATE 35]     SET
        { units                  [0]     SEQUENCE OF SEQUENCE {UNIT,
INTEGER },
        is-style-single          [1]     BOOLEAN }

QTY-WITH-UNITS   ::= [PRIVATE 36]     SEQUENCE
        {                                QTY,
                                         Q-WITH-UNITS }

MQTY-WITH-UNITS          ::= [PRIVATE 37]     SEQUENCE
        {                                MQTY,
                                         Q-WITH-UNITS }

MQTY-RANGE          ::= [PRIVATE 38]     SEQUENCE
        {                                Q-RANGE,
                                         MEASUREMENT}

QTY-RANGE ::= [PRIVATE 39]     SEQUENCE
        {                                Q-RANGE }

BULKY-DATA          ::= [PRIVATE 48]     SET
        {                                EHCR-INFO,
         logical-type            [0]     PLAIN-TEXT }

ELECTRONIC-DATA          ::= [PRIVATE 49]     SET
        {                                BULKY-DATA,
        is-reference             [0]     BOOLEAN,
        ref-electronic           [1]     URI             OPTIONAL,
        elec-data                [2]     BIT-REF         OPTIONAL }

PHYSICAL-DATA    ::= [PRIVATE 50]     SET
        {                                BULKY-DATA,
         storage-type            [0]     PLAIN-TEXT              OPTIONAL,
         storage-loc             [1]     OCTET STRING            OPTIONAL,
        reference                [2]     OCTET STRING}

ALIEN-DATA          ::= [PRIVATE 51]     SET
        {                                ELECTRONIC-DATA,
        method                   [0]     URI}

MULTIMEDIA-DATA          ::= [PRIVATE 52]     SET
        {                                ELECTRONIC-DATA,
        format                   [0]     PLAIN-TEXT,
        revision-id              [1]     OCTET STRING            OPTIONAL,
        size                     [2]     INTEGER                 OPTIONAL }
```

```
OCCASION    ::= [PRIVATE 53]      SET
        {                         EHCR-INFO}

DATE-RANGE          ::= [PRIVATE 54]      SET
        {                                 EHCR-INFO,
        min                 [0]           MOMENT,
        max                 [1]           MOMENT,
        incl-min            [2]           BOOLEAN,
        incl-max            [3]           BOOLEAN,
        within              [4]           BOOLEAN }

MOMENT      ::= [PRIVATE 55]      SET
        {                         OCCASION}

DATE-TIME   ::= [PRIVATE 56]      SET
        {                         MOMENT,
        dt-date             [0]   DATE,
        dt-time             [1]   TIME }

BOOL ::= [PRIVATE 57]     SET
        {                         EHCR-INFO,
        value               [0]   BOOLEAN }

UNIT  ::= [PRIVATE 58]     SET
        { term              [0]   TERM-REF,
          system            [1]   UNIT-SYSTEM       OPTIONAL }

UNIT-SYSTEM         ::= [PRIVATE 59]      SET
        { name              [0]   TERM-REF }

PROVIDER    ::= [PRIVATE 65]      SET
        {                         ID}

PERSON      ::= [PRIVATE 66]      SET
        {                         PROVIDER,
        name                [0]   PERSON-NAME,
        photo               [1]   BIT-REF       OPTIONAL,
        dt-photo-taken      [2]   DATE-TIME     OPTIONAL }

NON-PATIENT         ::= [PRIVATE 67]      SET
        {                                 PERSON,
        addresses                 [0]     SET {TYPED-ADDRESS } OPTIONAL,
        contact-nrs               [1]     SET {CONTACT-NR}OPTIONAL,
        net-addresses             [2]     SET    {NET-ADDRESS}OPTIONAL }
```

```
STAFF-MEMBER    ::= [PRIVATE 68]    SET
        {                                   NON-PATIENT,
         grade                  [0]     PLAIN-TEXT OPTIONAL,
         position               [1]     PLAIN-TEXT OPTIONAL }

HCP     ::= [PRIVATE 69]    SET
        {                                   STAFF-MEMBER,
         profession             [0]     PLAIN-TEXT OPTIONAL,
         reg                    [1]     HCP-REGISTRATION}

HCP-REGISTRATION        ::= [PRIVATE 70]    SET
        { profession            [0]     PLAIN-TEXT          OPTIONAL,
          regs                  [1]     SET {REGISTRATION}      }

TOOL ::= [PRIVATE 71]   SET
        {                               PROVIDER,
         name                   [0]     OCTET STRING,
        tool-type               [1]     TERM-REF    OPTIONAL}

PATIENT     ::= [PRIVATE 72]    SET
        {                               PERSON,
        date-of-birth           [0]     DATE-TIME,
        gender                  [1]     GENDER-CODE }

PERSON-NAME     ::= [PRIVATE 73]    SET
        {surname                [0]     OCTET STRING,
         forenames              [1]     SET OF OCTET STRING     OPTIONAL,
         titles                 [2]     MULTI-TEXT              OPTIONAL,
         letters                [3]     OCTET STRING            OPTIONAL}

ADDRESS     ::= [PRIVATE 74]    SET
        {addr-lines             [0]     SET {ADDRESS-LINE}OPTIONAL,
         postcode               [1]     OCTET STRING    OPTIONAL,
         valid-from             [2]     SET {DATE}      OPTIONAL,
         invalid-from           [3]     SET {DATE}      OPTIONAL}

HCF     ::= [PRIVATE 74]    SET
        { hcfid                 [0]     HCFID,
        name                    [1]     OCTET STRING,
        address                 [2]     ADDRESS         OPTIONAL,
        type-of-hcf             [3]     TERM-REF        OPTIONAL,
        reg                     [4]     REGISTRATION,
        contact-nrs             [5]     SET {CONTACT-NR}OPTIONAL,
        net-addresses           [6]     SET {NET-ADDRESS }OPTIONAL,
        photo                   [7]     BIT-REF         OPTIONAL,
```

```
        dt-photo-taken          [8]     DATE-TIME           OPTIONAL}

ADDRESS-LINE      ::= [PRIVATE 76]      SET
        {addr-line-type         [0]     TERM-REF            OPTIONAL,
         add-line-text          [1]     OCTET STRING        OPTIONAL}

NET-ADDRESS       ::= [PRIVATE 77]      SET
        {net-addr-type          [0]     TERM-REF,
         net-addr               [1]     URI                 OPTIONAL,
         comment                [2]     OCTET STRING        OPTIONAL,
         comment-lang           [3]     GEHR-LANG           OPTIONAL,
         valid-from             [4]     SET {DATE}          OPTIONAL,
         invalid-from           [5]     SET {DATE}          OPTIONAL}

REGISTRATION      ::= [PRIVATE 78]      SET
        {reg-country            [0]     COUNTRY-CODE,
         reg-type               [1]     TERM-REF,
         reg-number             [2]     OCTET STRING}

TYPED-ADDRESS     ::= [PRIVATE 79]      SET
        {addr-type              [0]     TERM-REF,
         address                [1]     ADDRESS     }

CONTACT-NR        ::= [PRIVATE 80]      SET
        {contact-nr-type        [0]     TERM-REF,
         number                 [1]     OCTET STRING,
         comment                [2]     OCTET STRING        OPTIONAL,
         comment-lang           [3]     GEHR-LANG           OPTIONAL,
         valid-from             [4]     SET {DATE}          OPTIONAL,
         invalid-from           [5]     SET  {DATE}         OPTIONAL}

VERSION     ::= [PRIVATE 81]      SET
        { revision              [0]     REVISION,
        dt-committed            [1]     DATE-TIME,
        hcp-authorising         [2]     POINTER-TO-HCP,
        hcp-legally-resp        [3]     POINTER-TO-HCP   OPTIONAL,
        recorder                [4]     POINTER-TO-STAFF-MEMBER
        OPTIONAL}

CODE-LINK  ::=[PRIVATE 82]      SET
        {concept-code           [0]     OCTET STRING        OPTIONAL,
         code-used              [1]     OCTET STRING        OPTIONAL }


GEF               ::= [PRIVATE 83] SET
```

```
{termsetinfo                [0]     SET OF CHOICE {TERMSET, LOCAL-
TERMSET}
 unitsystem                 [1]     SET OF UNITSYSTEM       OPTIONAL
 hcps                       [2]     SET OF HCP              OPTIONAL
 staffmembers               [3]     SET OF STAFF-MEMBER     OPTIONAL
 tools                      [4]     SET OF TOOL             OPTIONAL
 non-patient                [5]     SET OF NON-PATIENT      OPTIONAL
 hcfs                       [6]     SET OF HCF
 source                     [7]     SET OF SOURCE           OPTIONAL
 extract                    [8]     SET OF EHCR-EXTRACT     }

EMPH-LEVEL          ::= ENUMERATED {low(0), medium(1), high(2)}

GEHR-LANG           ::= ENUMERATED {en(0), fr(1), no(2)}

GENDER-CODE         ::= ENUMERATED {male(0), female(1), unknown(2)}

LEG-SOURCE-TYPE ::= ENUMERATED {paper(0), electronic(1), other(2)}

END
```

Appendix E – Legacy Intermediate Format

In this appendix the Legacy Intermediate Format (LIF) language is formally expressed in Extended Bacchus Naur Format. The language is shown in both structure order, page E1, and also Alphabetical order, page E10. See section 5.5 for a full explanation of the LIF.

LIF in EBNF presented here in Structure order.

LIF ::= LEGACY_SOURCE
LEGACY_SYSTEM_INFO
{TERMSET_INFO}
{UNIT_SYSTEM_INFO}
{HEALTH_CARE_PROFESSIONALS}
{STAFF_MEMBERS}
{TOOLS}
{NON_PATIENTS}
HEALTH_CARE_FACILITIES
{EHCR_SOURCES[3]}
PATIENTS
END LEGACY_SOURCE

LEGACY_SYSTEM_INFO ::=
legacy_name = STR
legacy_type = LEGACY_SOURCE_TYPE
owning_hcf[4] = HCF_LABEL

TERMSET_INFO ::= LOCAL_TEMRSET_INFO |
NON_LOCAL_TERMSET_INFO

NON_LOCAL_TERMSET_INFO ::= TERMSET ∇ TERMSET_LABEL [∇ DEFAULT[5] [∇ UNITS[6]]]
TERMSET_BODY
END TERMSET

LOCAL_TERMSET_INFO ::= LOCAL_TERMSET ∇ TERMSET_LABEL [∇ DEFAULT[7] [∇ UNITS[8]]]
LOCAL_TERMSET_BODY
END TERMSET

---

[3] EHCR = Electronic Health Care Record. EHCR_SOURCES is used to indicate, where appropriate, the source of data which originated from another EHCR source

[4] The Health Care Facility (HCF) from where the information in the LIF is being taken

[5] Only one term set may be defined to be the default term set

[6] Only one term set may be defined to be the default units term set.

[7] Only one term set may be defined to be the default term set

[8] Only one term set may be defined to be the default units term set.

```
UNIT_SYSTEM_INFO ::= UNIT_SYSTEM
                         name = TR
                         termset = TERMSET_LABEL
                     END UNIT_SYSTEM

PATIENTS ::=
        PATIENT_INFO
        { PATIENT_INFO }

HEALTH_CARE_PROFESSIONALS ::=
        HCP HCP_LABEL[9]
                HCP_BODY
        END HCP

STAFF_MEMBERS ::=
        STAFF_MEMBER   SM_LABEL[10]
                SM_BODY
        END STAFF_MEMBER

TOOLS[11] ::=
        TOOL TOOL_LABEL[12]
                TOOL_BODY
        END TOOL

NON_PATIENTS ::=
        NON_PATIENT  NP_LABEL[13]
                NP_BODY
        END NON_PATIENT

HEALTH_CARE_FACILITIES ::=
        HCF  HCF_LABEL[14]
                HCF_BODY
        END HCF
        {HCF  HCF_LABEL
                HCF_BODY
        END HCF}
```

---

[9] Each HCP label must be unique within one LIF

[10] Each staff member label must be unique within one LIF

[11] Used to indicate a sofware tool which has been used to generate a piece of information in the record

[12] Each non patient label must be unique within one LIF

[13] Each HCF label must be unique within one LIF

[14] Each source label must be unique within one LIF

```
EHCR_SOURCES ::=
        SOURCE  SOURCE_LABEL[15]
                SOURCE_BODY
        END SOURCE

STR ::= " { CHAR } "
LEGACY_SOURCE_TYPE ::= P | E[16]
HCF_LABEL ::= LABEL

TERMSET_LABEL ::= LABEL
TERMSET_BODY[17] ::=        code = STR
                        [name = STR
                        revision = STR
                        reg_agency = STR]

LOCAL_TERMSET_BODY ::=      [code = STR]
                            name = STR
                            revision = STR
                            reg_agency = STR
                            [reg_agency_source = SOURCE_LABEL][18]

PATIENT_INFO ::=
        EHCR_EXTRACT
                EXTRACT_ATTRIBUTES
                SUBJECT_OF_EXTRACT
                ADMIN_TRAN | ADMIN_SUMMARY_TRAN
                {GENERAL_TRANSACTION }
        END EHCR_EXTRACT

HCP_LABEL ::= LABEL
HCP_BODY ::=        regs = REGISTRATION {, REGISTRATION}
                    [profession = PT]
                    SM_BODY

SM_LABEL ::= LABEL
SM_BODY ::=         [grade = PT]
                    [position = PT]
                    NP_BODY
```

[15] Each label must be unique within one LIF

[16] P signifies paper records. E signifies electronic records.

[17] For recognised term sets, the identifying code is mandatory

[18] If this attribute is not present, the source is assumed to be the same as the source of this LIF.

```
TOOL_LABEL ::= LABEL
TOOL_BODY ::= name = STR
              [tool_type = TR]

NP_LABEL ::= LABEL
NP_BODY ::= [addresses = TYPED_ADDRESS {, TYPED_ADDRESS}]
            [contact_nrs = CONTACT_NO {, CONTACT_NO}]
            [net_addrs = NET_ADDR {, NET_ADDR}]
            PERSON_NAME
            [photo = DT, FNSTR]

HCF_BODY ::=      name = STR
            [ADDRESS_INFO]
            [type_of_hcf = TR]
            reg = REGISTRATION
            [contact_nrs = CONTACT_NO{,CONTACT_NO}]
            [net_addrs = NET_ADDR{,NET_ADDR}]
            [photo = DT, FNSTR]

SOURCE_LABEL ::= LABEL
SOURCE_BODY ::= name = STR
                [net_addrs = NET_ADDR {, NET_ADDR}]
                [revsion_id = STR]
                [origin = STR]
                [owning_hcf = HCF_LABEL]
                [legacy_name = STR
                legacy_type = LEGACY_SOURCE_TYPE]

CHAR ::= a - z | A - Z | 0 - 9 | _ {a - z | A - Z | 0 - 9 | _ }[19]
LABEL ::= CHAR { CHAR}

EXTRACT_ATTRIBUTES ::=      [ehcr_id[20] = STR]
                            [dt_creation = DT]
                            [hcp_created_by = LABEL]
SUBJECT_OF_EXTRACT ::=
            SUBJECT
```

---

[19] A range in a terminal definition is denoted by a hyphen eg a - z denotes lowercase alphabetic characters

[20] if there exists a unique id on the legacy system, it could be used here and could later be used to link back if further information is extracted at a later date. Note however, that a legacy id on its own is insufficient to guarantee identification of a patient and should only ever be used in conjunction with other identification information. If no id is provided, the LIF → GEF compiler will provide one.

(VERSION[21]
VERSION_BODY
END VERSION
{VERSION
VERSION_BODY
END VERSION}) |
VERSION_BODY
END SUBJECT

VERSION_BODY ::= PERSON NAME
PERSON_IDENTIFICATION
[hcp_authorising = LABEL]
[hcp_legally_resp = LABEL]
[recorder = LABEL]
[dt_committed = DT]

ADMIN_TRAN ::= ADMIN
TRANSACTION_BODY
END ADMIN

ADMIN_SUMMARY_TRAN ::= ADMIN_SUMMARY
[period = DR]
TRANSACTION_BODY
END ADMIN_SUMMARY

GENERAL_TRANSACTION ::= CONTACT_TRAN | SUMMARY_TRAN |
ADMIN_TRAN | REPORT_TRAN |
CONT_CARE_TRAN | NOTA_BENE_TRAN |
ADMIN_SUMMARY_TRAN

REGISTRATION ::= COUNTRY_CODE , TR , STR
PT ::= QTR | (STR [: LANG]) | POSSTR
TR ::= CODED_TR | TEXT_TR
TYPED_ADDRESS ::= TR : ADDRESS_INFO
CONTACT_NO ::= TR : STR [, STR [: LANG]]
NET_ADDR ::= TR : URI [, STR [: LANG]]
PERSON_NAME ::= [TITLES = MT]
name = [STR {, STR } ,] STR
[letters = STR]
PERSON_IDENTIFICATION ::= date_of_birth = DT
gender = GENDER_CODE
[photo = DT , FNSTR]

---

[21] If a legacy system only holds one version of subject, then VERSION and END VERSION may be omitted. In this case, version 1.0 will be assigned. If the legacy source does hold more than one version, then VERSION and END VERSION must be used and must be in order - oldest first.

```
ADDRESS_INFO ::=        ADDRESS
                        addr_lines = ADDR_LINE {, ADDR_LINE}
                        [postcode = STR]
                        [valid_from = DATE {, DATE}]
                        [invalid_from =  DATE {, DATE}]
                    END ADDRESS

TRANSACTION_BODY ::=        [access_rights = PERMS]
                        [amend_rights = PERMS]
                        [dt_commited = DT]
                        [hcp_authorising = LABEL]
                        [hcp_legally_resp = LABEL]
                        [recorder = LABEL]
                        {OBSERVATION
ANNOTATED_OBSERVATIONS}
DR ::= [NOT] ([ FROM_PART ] TO_PART ) | FROM_PART
CONTACT_TRAN ::=        CONTACT [ TRANS_LABEL]
                        [dt_occured = DT]
                        [contact_with = HCP_LABEL]
                        TRANSACTION_BODY
                    END CONTACT
SUMMARY_TRAN ::=        SUMMARY [ TRANS_LABEL]
                        [period = DR]
                        TRANSACTION_BODY
                    END SUMMARY
REPORT_TRAN ::=  REPORT [ TRANS_LABEL]
                        [in_reply_to        =        LINKABLE_LABEL
{,LINKABLE_LABEL}]
                        TRANSACTION_BODY
                    END REPORT
CONT_CARE_TRAN ::=      CONT_CARE[ TRANS_LABEL]
                        [period = DR]
                        TRANSACTION_BODY
                    END CONT_CARE
NOTA_BENE_TRAN ::=  NOTA_BENE
                        TRANSACTION_BODY
                    END NOTA_BENE

COUNTRY_CODE ::= uk

QTR ::= TR [ ( TRQ {, TRQ } ) ]
LANG ::= UKeng[22]
POSSTR ::= < STR : TERMSET_LABEL {, TERMSET_LABEL } >
CODED_TR ::= < <  ( CODE_USED [, CONCEPT_CODE] ) | CONCEPT_CODE >
:                       TERMSET_LABEL >
```

[22] Further countries would be added to the set for non UK systems.

TEXT_TR ::= < STR [: TERMSET_LABEL { , TERMSET_LABEL} [+]] > [: LANG]

URI ::= STR
MT ::= PT {, PT }
DT ::= dd-mm-yyyy∇ hh:mm:ss | dd-mm-yyyy | hh:mm:ss
GENDER_CODE ::= F | M | U | -
FNSTR ::= STR

ADDR_LINE ::= [TR] STR
DATE ::= dd-mm-yyyy
PERMS ::= 1 | 2 | 4 | 8 | 16 | 32[23]
OBSERVATION ::= ITEM | COLLECTION
ANNOTATED_OBSERVATIONS ::= HEADING ENTRY_NAME
[emphasis = EMPH_LEVEL]
[recorder = HCF_LABEL | SM_LABEL]
OBSERVATION |
ANNOTATED_OBSERVATIONS
{ OBSERVATION |
ANNOTATED_OBSERVATIONS }
END HEADING

FROM_PART ::= FROM DT [INC]
TO_PART ::= TO DT [INC]
TRANS_LABEL ::= LABEL
LINKABLE_LABEL ::= OBS_LABEL | TRANS_LABEL
TRQ ::= TR
CODE_USED ::= STR
CONCEPT_CODE ::= STR
ITEM ::= [derived∇] HRI ∇ ENTRY_NAME [∇ OBS_LABEL]
OBS_ATTRS
[ct_comment = MT]
[dt_OBSERVED = DT]
[certainty = PT]
[ct_emphasis = EMPH_LEVEL]
[content = EHCR_INFO]
END∇ HRI

---

[23] patient = 1

HCP legally responsible = 2

HCP authorising = 4

any HCP = 8

other staff = 16

other = 32

```
COLLECTION ::= HRI_COLLECTION ∇ ENTRY_NAME [∇ OBS_LABEL]
                        OBS_ATTRS
                        OBSERVATION | ANNOTATED_OBSERVATION
                        { OBSERVATION | ANNOTATED_OBSERVATION }
                    ENDV HRI_COLLECTION
ENTRY_NAME ::= TR
EMPH_LEVEL ::= L | M | H
OBS_LABEL ::= LABEL
OBS_ATTRS ::=       [emphasis = EMPH_LEVEL ]
                    [recorder = SM_LABEL ]
                    [links = LINKABLE_LABEL { , LINKABLE_LABEL } ]
                    [access_rights = PERMS ]
                    [cx_comment = MT ]
                    [in_reply_to = LINKABLE_LABEL  { , LINKABLE_LABEL } ]
                    [info_provider = PROVIDER_LABEL ]

EHCR_INFO ::= BOOL | OCCASION | MT | PT | QUANTITY | QTY_RATIO |
QTY_RANGE
| MQTY_RANGE | BULKY_DATA
PROVIDER_LABEL  ::=  SM_LABEL  |  HCP_LABEL  |  NP_LABEL  |
TOOL_LABEL

BOOL ::= TRUE | FALSE
OCCASION ::= DR | DT
QUANTITY ::= QTY | MQTY
QTY_RATIO ::= QUANTITY [∇ OF∇ TR] ∇ PER∇ QUANTITY [∇ OF∇ TR]
Q_RANGE ::= [NOT] ([FROMQPART] TOQPART) | FROMQPART
QTY_RANGE ::= Q_RANGE
MQTY_RANGE ::= Q_RANGE ON ∇ PT[TO∇ STD_NUMERIC[%]]
BULKY_DATA ::= PHYSICAL_DATA | ALIEN_DATA | MM_DATA
QTY ::=      ((STD_NUMERIC  [%]  )  |  (STD_NUMERIC∇  UNIT_TR  [^
INTEGER]
      {. UNIT_TR [^ INTEGER] } | UNIT_TR {∇ UNSIGNED_INTEGER∇
UNIT_TR }
      ∇ UNSIGNED_STD_NUMERIC UNIT_TR)) [UNSIGNED_INTEGER SF |
DP]
MQTY ::= Measured QTY ON∇ PT [TO∇ STD_NUMERIC [%]]
FROMQPART ::= FROM QTY [INC]
TOQPART ::= TO QTY [INC]
PHYSICAL_DATA ::= PT∇ IS∇ PHYS[PT,]IN [STR ,]REF STR
ALIEN_DATA ::= PT∇ IS∇ ALIEN[INC∇][URI ,] REF FNSTR∇ BY∇ URI
MM_DATA ::= PT∇ IS∇ MM[INC∇] [URI ,] REF FNSTR∇ AS∇ PT [, STR]
STD_NUMERIC ::= [-] UNSIGNED_STD_NUMERIC
UNIT_TR ::= TR
INTEGER ::= [-] UNSIGNED_INTEGER
UNSIGNED_INTEGER ::= 0 - 9 {0 - 9}
```

UNSIGNED_STD_NUMERIC ::= UNSIGNED_INTEGER | UNSIGNED_REAL
REAL ::= [-] UNSIGNED_REAL
UNSIGNED_REAL ::= UNSIGNED_INTEGER . UNSIGNED_INTEGER |
(UNSIGNED_INTEGER E [+ | -] UNSIGNED_INTEGER)

The Elements of the LIF presented in alphabetical order:

ADDR_LINE ::= [TR] STR

ADDRESS_INFO ::= ADDRESS
addr_lines = ADDR_LINE {, ADDR_LINE}
[postcode = STR]
[valid_from = DATE {, DATE}]
[invalid_from = DATE {, DATE}]
END ADDRESS

ADMIN_TRAN ::= ADMIN
TRANSACTION_BODY
END ADMIN

ADMIN_SUMMARY_TRAN ::= ADMIN_SUMMARY
[period = DR]
TRANSACTION_BODY
END ADMIN_SUMMARY

ALIEN_DATA ::= PTV ISV ALIEN[INCV][URI ,] REF FNSTRV BYV URI

ANNOTATED_OBSERVATIONS ::= HEADING ENTRY_NAME
[emphasis = EMPH_LEVEL]
[recorder = HCF_LABEL | SM_LABEL]
OBSERVATION |
ANNOTATED_OBSERVATIONS
{ OBSERVATION |
ANNOTATED_OBSERVATIONS }
END HEADING

BOOL ::= TRUE | FALSE

BULKY_DATA ::= PHYSICAL_DATA | ALIEN_DATA | MM_DATA

CHAR ::= a - z | A - Z | 0 - 9 | _ {a - z | A - Z | 0 - 9 | _ }[24]

CODE_USED ::= STR

CODED_TR ::= << (CODE_USED [, CONCEPT_CODE]) | CONCEPT_CODE >
: TERMSET_LABEL >

COLLECTION ::= HRI_COLLECTION ∇ ENTRY_NAME [∇ OBS_LABEL]
OBS_ATTRS

---

[24] A range in a terminal definition is denoted by a hyphen eg a - z denotes lowercase alphabetic characters

OBSERVATION | ANNOTATED_OBSERVATION
{ OBSERVATION | ANNOTATED_OBSERVATION }
ENDV HRI_COLLECTION

CONCEPT_CODE ::= STR

CONT_CARE_TRAN ::= CONT_CARE[TRANS_LABEL]
[period = DR]
TRANSACTION_BODY
END CONT_CARE

CONTACT_NO ::= TR : STR [, STR [: LANG]]

CONTACT_TRAN ::= CONTACT [TRANS_LABEL]
[dt_occured = DT]
[contact_with = HCP_LABEL]
TRANSACTION_BODY
END CONTACT

COUNTRY_CODE ::= uk

DATE ::= dd-mm-yyyy

DR ::= [NOT] ([FROM_PART] TO_PART) | FROM_PART

DT ::= dd-mm-yyyyV hh:mm:ss | dd-mm-yyyy | hh:mm:ss

EHCR_INFO ::= BOOL | OCCASION | MT | PT | QUANTITY | QTY_RATIO | QTY_RANGE
| MQTY_RANGE | BULKY_DATA

EHCR_SOURCES ::=
SOURCE SOURCE_LABEL[25]
SOURCE_BODY
END SOURCE

ENTRY_NAME ::= TR

EMPH_LEVEL ::= L | M | H

EXTRACT_ATTRIBUTES ::= [ehcr_id[26] = STR]

---

[25] Each label must be unique within one *LIF*

[26] if there exists a unique id on the legacy system, it could be used here and could later be used to link back if further information is extracted at a later date. Note however, that a legacy id on its own is insufficient to guarantee identification of a patient and

```
                    [dt_creation = DT]
                    [hcp_created_by = LABEL]

FNSTR ::= STR

FROM_PART ::= FROM DT [INC]

FROMQPART ::= FROM QTY [INC]

GENDER_CODE ::= F | M | U | -

GENERAL_TRANSACTION ::=   CONTACT_TRAN  |  SUMMARY_TRAN  |
ADMIN_TRAN |                              REPORT_TRAN      |
CONT_CARE_TRAN | NOTA_BENE_TRAN |
ADMIN_SUMMARY_TRAN

HCF_BODY ::=      name = STR
                  [ADDRESS_INFO]
                  [type_of_hcf = TR]
                  reg = REGISTRATION
                  [contact_nrs = CONTACT_NO{,CONTACT_NO}]
                  [net_addrs = NET_ADDR{,NET_ADDR}]
                  [photo = DT, FNSTR]

HCF_LABEL ::= LABEL

HCP_BODY ::=      regs = REGISTRATION {, REGISTRATION}
                  [profession = PT]
                  SM_BODY

HCP_LABEL ::= LABEL

HEALTH_CARE_FACILITIES ::=
      HCF  HCF_LABEL[27]
            HCF_BODY
      END HCF
      {HCF  HCF_LABEL
            HCF_BODY
      END HCF}

HEALTH_CARE_PROFESSIONALS ::=
```

---

should only ever be used in conjunction with other identification information. If no id is provided, the LIF → GEF compiler will provide one.

[27] Each source label must be unique within one LIF

HCP HCP_LABEL[28]
HCP_BODY
END HCP

INTEGER ::= [-] UNSIGNED_INTEGER

ITEM ::= [derived∇] HRI ∇ ENTRY_NAME [∇ OBS_LABEL]
OBS_ATTRS
[ct_comment = MT]
[dt_OBSERVED = DT]
[certainty = PT]
[ct_emphasis = EMPH_LEVEL]
[content = EHCR_INFO]
END∇ HRI

LABEL ::= CHAR { CHAR}

LANG ::= UKeng[29]

LEGACY_SOURCE_TYPE ::= P | E[30]

LEGACY_SYSTEM_INFO ::=
legacy_name = STR
legacy_type = LEGACY_SOURCE_TYPE
owning_hcf[31] = HCF_LABEL

LIF ::= LEGACY_SOURCE
LEGACY_SYSTEM_INFO
{TERMSET_INFO}
{UNIT_SYSTEM_INFO}
{HEALTH_CARE_PROFESSIONALS}
{STAFF_MEMBERS}
{TOOLS}
{NON_PATIENTS}
HEALTH_CARE_FACILITIES
{EHCR_SOURCES[32]}

---

[28] Each HCP label must be unique within one LIF

[29] Further countries would be added to the set for non UK systems.

[30] P signifies paper records. E signifies electronic records.

[31] The Health Care Facility (HCF) from where the information in the LIF is being taken

[32] EHCR = Electronic Health Care Record. EHCR_SOURCES is used to indicate, where appropriate, the source of data which originated from another EHCR source

PATIENTS
END LEGACY_SOURCE

LINKABLE_LABEL ::= OBS_LABEL | TRANS_LABEL

LOCAL_TERMSET_BODY ::= [code = STR]
name = STR
revision = STR
reg_agency = STR
[reg_agency_source = SOURCE_LABEL][33]

LOCAL_TERMSET_INFO ::=
LOCAL_TERMSET ∇ TERMSET_LABEL [∇ DEFAULT[34] [∇ UNITS[35]]]
LOCAL_TERMSET_BODY
END TERMSET

MM_DATA ::= PT∇ IS∇ MM[INC∇] [URI ,] REF FNSTR∇ AS∇ PT [, STR]

MQTY ::= Measured QTY ON∇ PT [TO∇ STD_NUMERIC [%]]

MQTY_RANGE ::= Q_RANGE ON ∇ PT[TO∇ STD_NUMERIC[%]]

MT ::= PT {, PT }

NET_ADDR ::= TR : URI [, STR [: LANG]]

NON_LOCAL_TERMSET_INFO ::=
TERMSET ∇ TERMSET_LABEL [∇ DEFAULT[36] [∇ UNITS[37]]]
TERMSET_BODY
END TERMSET

NON_PATIENTS ::=
NON_PATIENT NP_LABEL[38]
NP_BODY
END NON_PATIENT

NOTA_BENE_TRAN ::= NOTA_BENE

---

[33] If this attribute is not present, the source is assumed to be the same as the source of this LIF.

[34] Only one term set may be defined to be the default term set

[35] Only one term set may be defined to be the default units term set.

[36] Only one term set may be defined to be the default term set

[37] Only one term set may be defined to be the default units term set.

[38] Each HCF label must be unique within one LIF

TRANSACTION_BODY
END NOTA_BENE

NP_BODY ::= [addresses = TYPED_ADDRESS {, TYPED_ADDRESS}]
[contact_nrs = CONTACT_NO {, CONTACT_NO}]
[net_addrs = NET_ADDR {, NET_ADDR}]
PERSON_NAME
[photo = DT, FNSTR]

NP_LABEL ::= LABEL

OBS_ATTRS ::= [emphasis = EMPH_LEVEL]
[recorder = SM_LABEL]
[links = LINKABLE_LABEL { , LINKABLE_LABEL }]
[access_rights = PERMS]
[cx_comment = MT]
[in_reply_to = LINKABLE_LABEL { , LINKABLE_LABEL }]
[info_provider = PROVIDER_LABEL]

OBS_LABEL ::= LABEL

OBSERVATION ::= ITEM | COLLECTION

OCCASION ::= DR | DT

PATIENTS ::=
PATIENT_INFO
{ PATIENT_INFO }

PATIENT_INFO ::=
EHCR_EXTRACT
EXTRACT_ATTRIBUTES
SUBJECT_OF_EXTRACT
ADMIN_TRAN | ADMIN_SUMMARY_TRAN
{GENERAL_TRANSACTION }
END EHCR_EXTRACT

PERMS ::= 1 | 2 | 4 | 8 | 16 | 32[39]

[39] patient = 1
HCP legally responsible = 2
HCP authorising = 4
any HCP = 8
other staff = 16
other = 32

```
PERSON_IDENTIFICATION ::=     date_of_birth = DT
                              gender = GENDER_CODE
                              [photo = DT , FNSTR ]

PERSON_NAME ::= [TITLES = MT]
                name = [ STR {, STR } , ] STR
                [letters = STR ]

PHYSICAL_DATA ::= PTV ISV PHYS[PT,]IN [STR ,]REF STR

POSSTR ::= < STR : TERMSET_LABEL {, TERMSET_LABEL } >

PROVIDER_LABEL ::= SM_LABEL | HCP_LABEL | NP_LABEL |
TOOL_LABEL

PT ::= QTR | (STR [: LANG ] ) | POSSTR

Q_RANGE ::= [NOT] ([FROMQPART] TOQPART) | FROMQPART

QTR ::= TR [ ( TRQ {, TRQ } ) ]

QTY ::=     ((STD_NUMERIC [%] ) | (STD_NUMERICV UNIT_TR [^
INTEGER]
      {. UNIT_TR [^ INTEGER] } | UNIT_TR {V UNSIGNED_INTEGERV
UNIT_TR }
      V UNSIGNED_STD_NUMERIC UNIT_TR)) [UNSIGNED_INTEGER SF |
DP]

QTY_RATIO ::= QUANTITY [V OFV TR] V PERV QUANTITY [V OFV TR]

QTY_RANGE ::= Q_RANGE

QUANTITY ::= QTY | MQTY

REAL ::= [-] UNSIGNED_REAL

REGISTRATION ::= COUNTRY_CODE , TR , STR

REPORT_TRAN ::= REPORT [ TRANS_LABEL]
                [in_reply_to = LINKABLE_LABEL
{,LINKABLE_LABEL}]
                TRANSACTION_BODY
                END REPORT

SOURCE_BODY ::= name = STR
```

```
                    [net_addrs = NET_ADDR {, NET_ADDR}]
                    [revsion_id = STR]
                    [origin = STR]
                    [owning_hcf = HCF_LABEL]
                    [legacy_name = STR
                    legacy_type = LEGACY_SOURCE_TYPE]
SOURCE_LABEL ::= LABEL

SM_BODY ::=         [grade = PT]
                    [position = PT]
                    NP_BODY

SM_LABEL ::= LABEL

STAFF_MEMBERS ::=
        STAFF_MEMBER   SM_LABEL[40]
                SM_BODY
        END STAFF_MEMBER

STD_NUMERIC ::= [-] UNSIGNED_STD_NUMERIC

STR ::= " { CHAR } "

SUBJECT_OF_EXTRACT ::=
                SUBJECT
                  (VERSION[41]
                      VERSION_BODY
                   END VERSION
                   {VERSION
                      VERSION_BODY
                    END VERSION}) |
                    VERSION_BODY
                END SUBJECT

SUMMARY_TRAN ::=        SUMMARY [ TRANS_LABEL]
                        [period = DR]
                        TRANSACTION_BODY
                    END SUMMARY

TERMSET_BODY[42] ::=    code = STR
```

[40] Each staff member label must be unique within one LIF

[41] If a legacy system only holds one version of subject, then VERSION END VERSION may be omitted. In this case, version 1.0 will be assigned. If the legacy source does holds more than one version, then VERSION END VERSION must be used and must be in order - oldest first.

[name = STR
revision = STR
reg_agency = STR]

TERMSET_INFO ::= LOCAL_TEMRSET_INFO |
NON_LOCAL_TERMSET_INFO

TERMSET_LABEL ::= LABEL

TEXT_TR ::= < STR [: TERMSET_LABEL { , TERMSET_LABEL} [+]] > [:
LANG]

TO_PART ::= TO DT [INC]

TOOL_BODY ::= name = STR
[tool_type = TR]

TOOL_LABEL ::= LABEL

TOOLS[43] ::=
TOOL TOOL_LABEL[44]
TOOL_BODY
END TOOL

TOQPART ::= TO QTY [INC]

TR ::= CODED_TR | TEXT_TR

TRANS_LABEL ::= LABEL

TRANSACTION_BODY ::= [access_rights = PERMS]
[amend_rights = PERMS]
[dt_commited = DT]
[hcp_authorising = LABEL]
[hcp_legally_resp = LABEL]
[recorder = LABEL]
{OBSERVATION |
ANNOTATED_OBSERVATIONS}

TRQ ::= TR

---

[42] For recognised term sets, the identifying code is mandatory

[43] Used to indicate a sofware tool which has been used to generate a piece of information in the record

[44] Each non patient label must be unique within one LIF

```
TYPED_ADDRESS ::= TR : ADDRESS_INFO

UNIT_SYSTEM_INFO ::= UNIT_SYSTEM
                         name = TR
                         termset = TERMSET_LABEL
                     END UNIT_SYSTEM

UNIT_TR ::= TR

UNSIGNED_INTEGER ::= 0 - 9 {0 - 9}

UNSIGNED_REAL  ::=  UNSIGNED_INTEGER  .  UNSIGNED_INTEGER  |
(UNSIGNED_INTEGER E       [+ | -] UNSIGNED_INTEGER)

UNSIGNED_STD_NUMERIC ::= UNSIGNED_INTEGER | UNSIGNED_REAL

URI ::= STR

VERSION_BODY ::=        PERSON NAME
                    PERSON_IDENTIFICATION
                    [hcp_authorising = LABEL]
                    [hcp_legally_resp = LABEL]
                    [recorder = LABEL]
                    [dt_committed = DT]
```

Appendix F - State Chart Diagram

This appendix presents the state chart diagram on which the CLIF compiler software was based. An oval shape represents each state; if a state has a sub-state this is represented by a smaller oval within the larger one. Some ovals remain empty which represents a collection point of possible routes that may be taken and are used to make the diagram less complicated. For a full explanation of the state chart diagram see section 6.3.2.1. All diagrams after the top-level diagram are in alphabetical order.

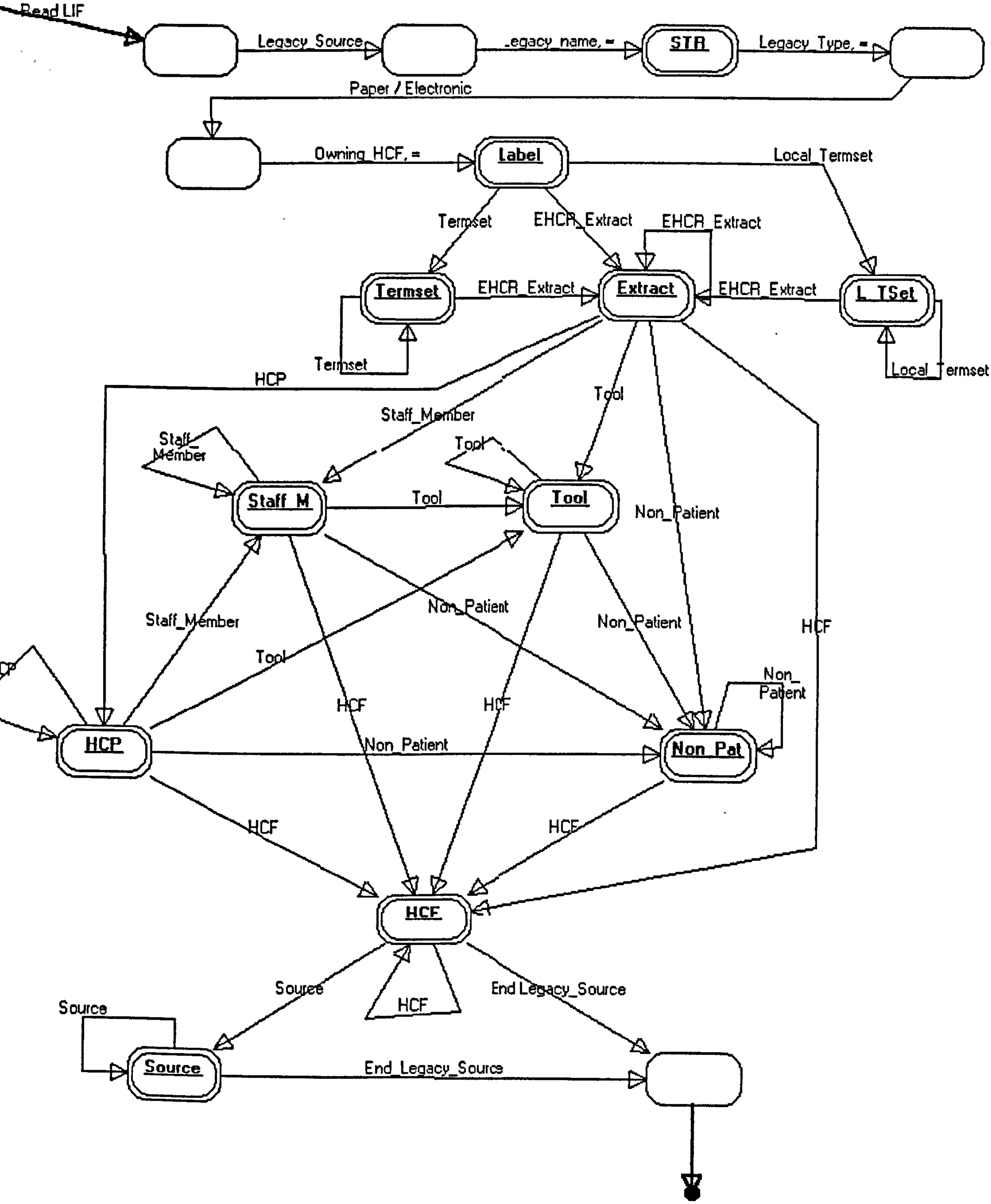


Diagram
Legacy Intermediate
Format – Top Level

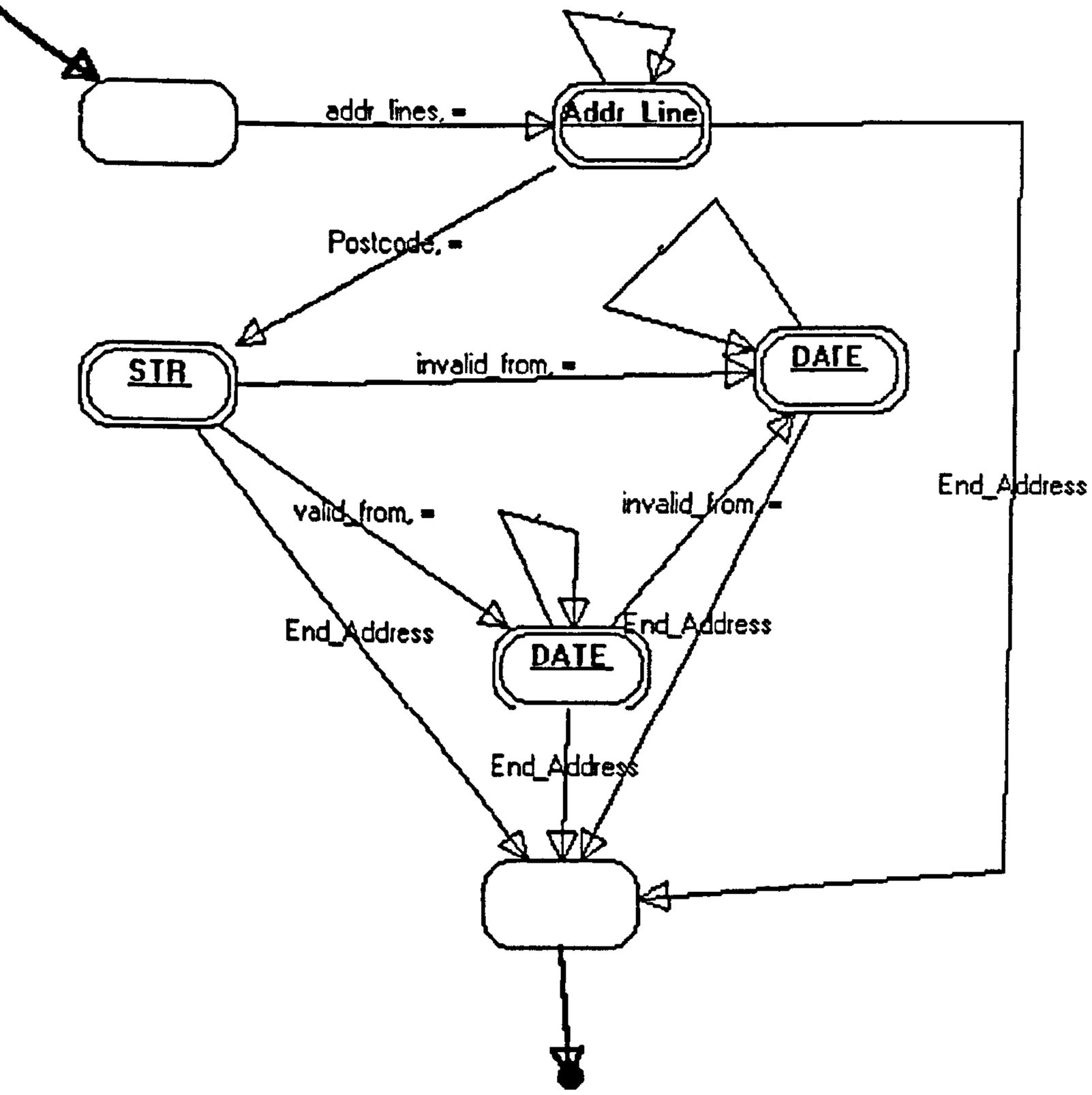
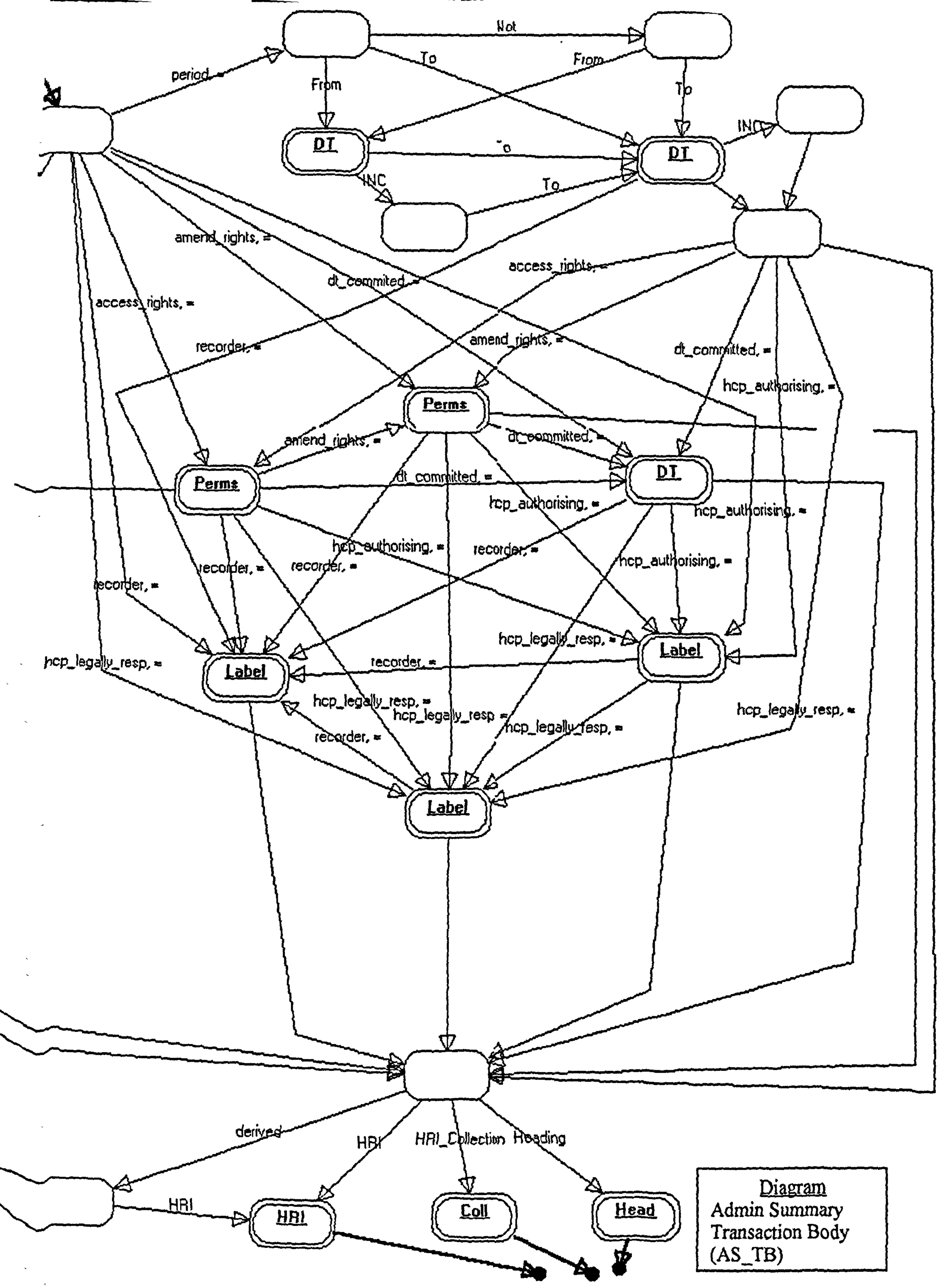


Diagram
Address Information
(Add_Info)

F2

Diagram
Address Line
(Addr_Line)

F3


Not
To
From
period, =
From
To
DI
To
DI
IND
INC
To
amend_rights, =
dt_commited, =
access_rights, =
access_rights, =
recorder, =
amend_rights, =
dt_committed, =
hcp_authorising, =
Perms
amend_rights, =
dt_committed, =
Perms
dt_committed, =
DI
hcp_authorising, =
hcp_authorising, =
hcp_authorising, =
recorder, =
recorder, =
recorder, =
hcp_authorising, =
recorder, =
hcp_legally_resp, =
Label
hcp_legally_resp, =
recorder, =
hcp_legally_resp, =
hcp_legally_resp =
hcp_legally_resp, =
hcp_legally_resp, =
recorder, =
Label
derived
HRI
HRI_Collection
Heading
HRI
HRI
Coll
Head
Diagram
Admin Summary
Transaction Body
(AS_TB)

F4

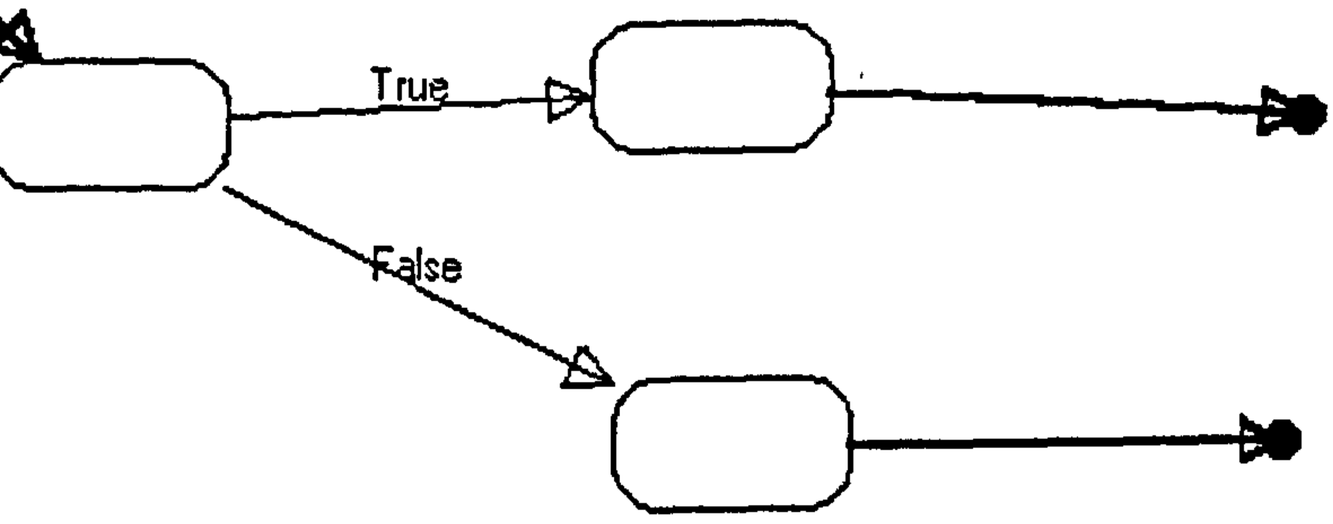


Diagram
Boolean
(Bool)

F5

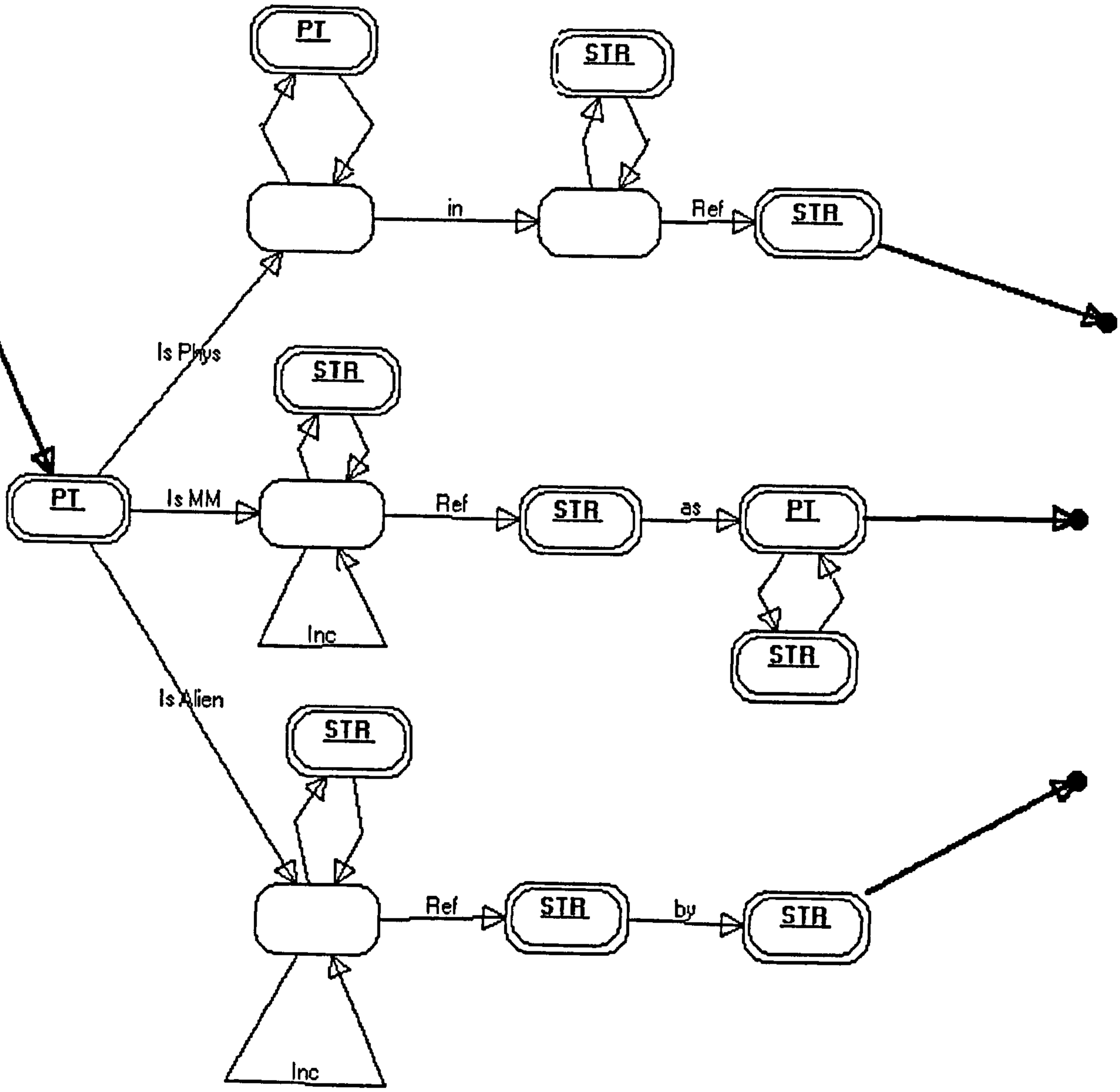


Diagram
Bulky Data

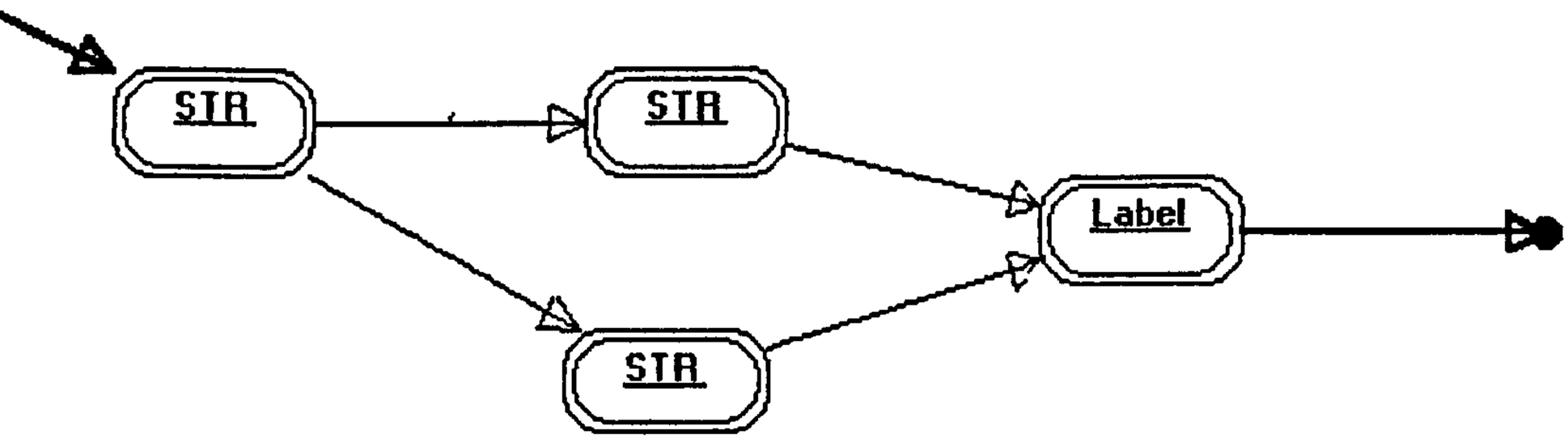
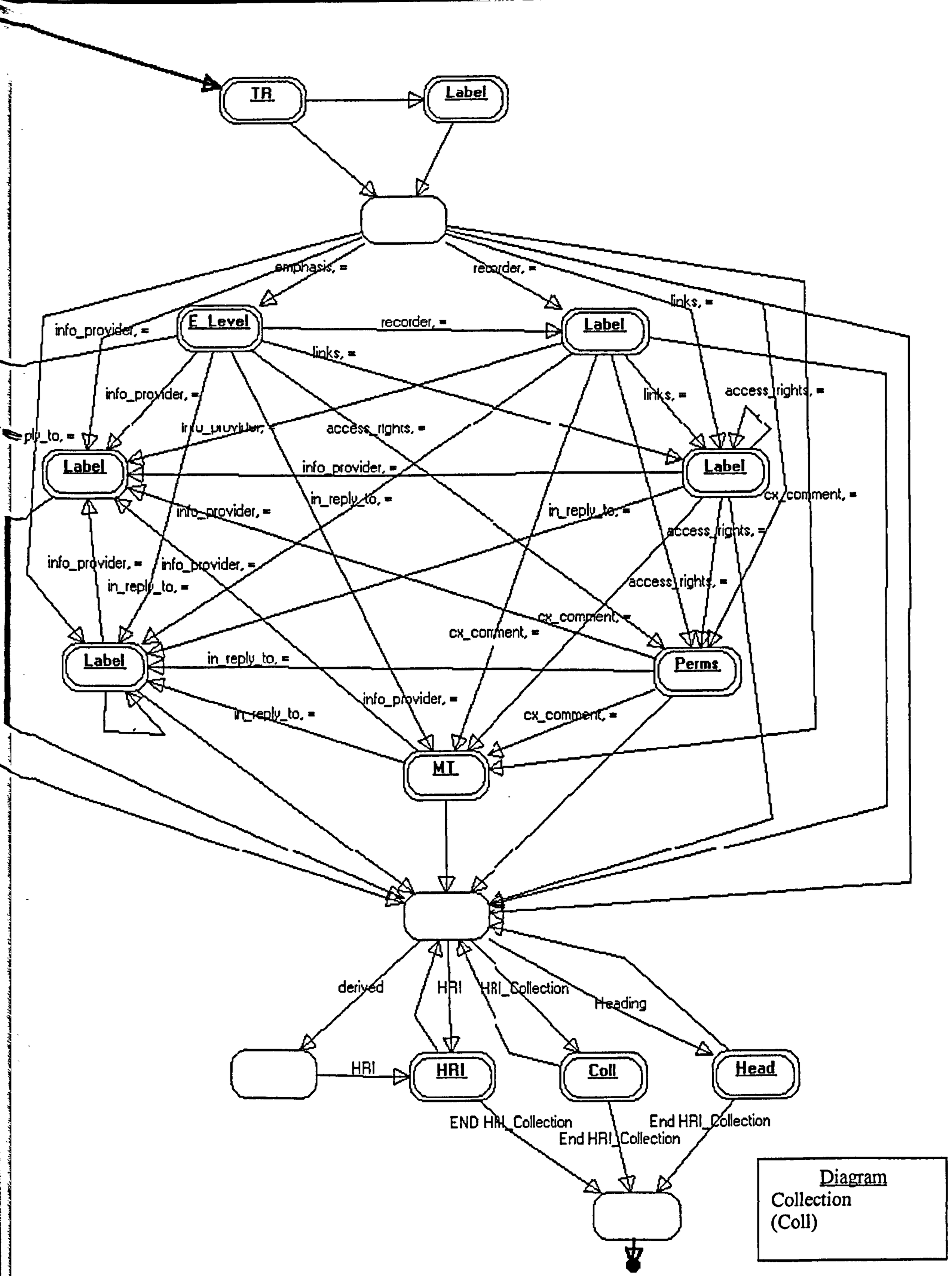


Diagram
Coded Term
Reference
(Coded_TR)

F7


TR
Label
emphasis, =
recorder, =
links, =
E_Level
info_provider, =
recorder, =
links, =
Label
info_provider, =
links, =
access_rights, =
Label
info_provider, =
in_reply_to, =
cx_comment, =
info_provider, =
in_reply_to, =
access_rights, =
info_provider, =
info_provider, =
in_reply_to, =
access_rights, =
cx_comment, =
cx_comment, =
Label
in_reply_to, =
Perms
info_provider, =
in_reply_to, =
cx_comment, =
MI
derived
HRI
HRI_Collection
Heading
HRI
HRI
Coll
Head
END HRI_Collection
End HRI_Collection
End HRI_Collection
Diagram
Collection
(Coll)

F8

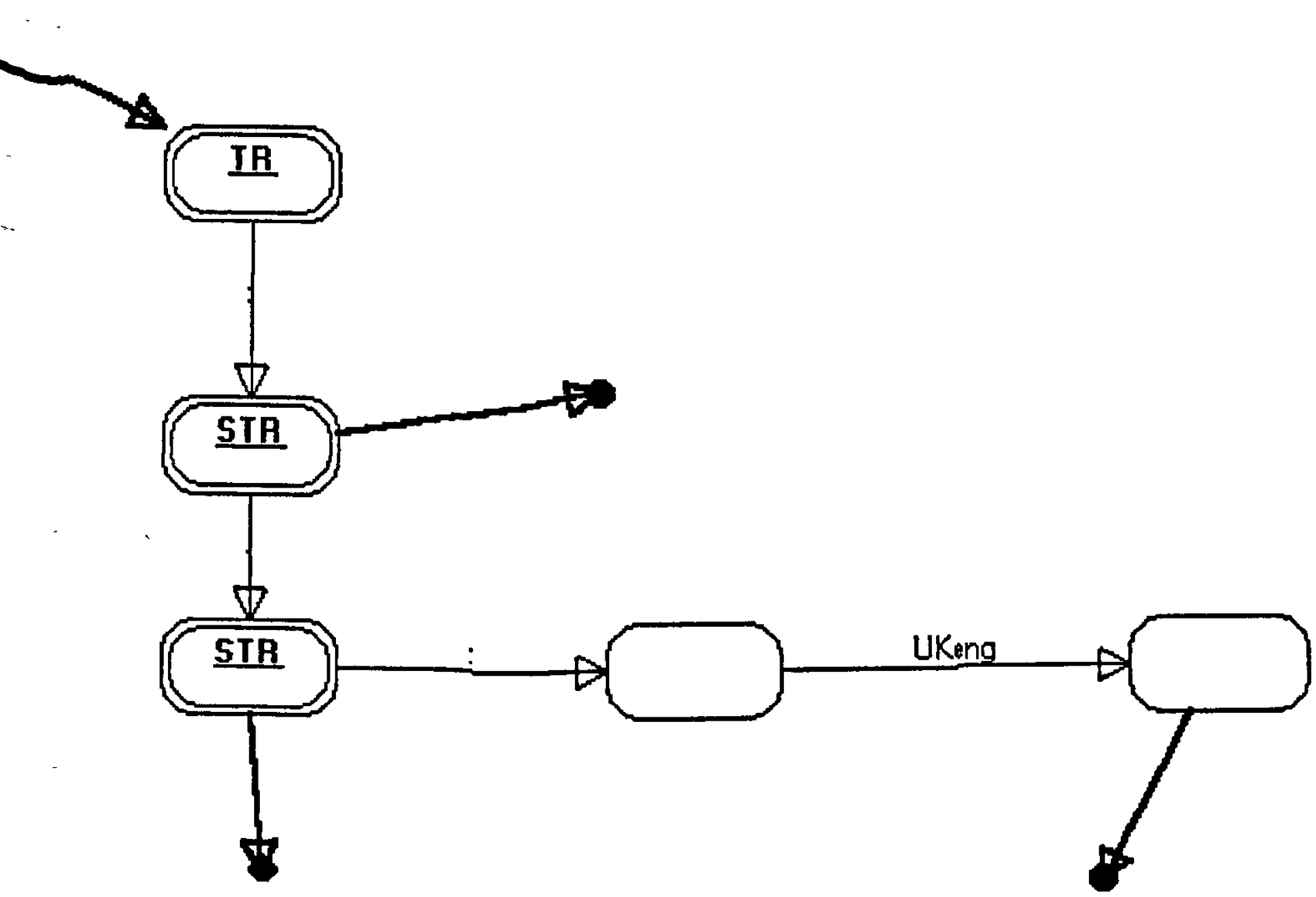
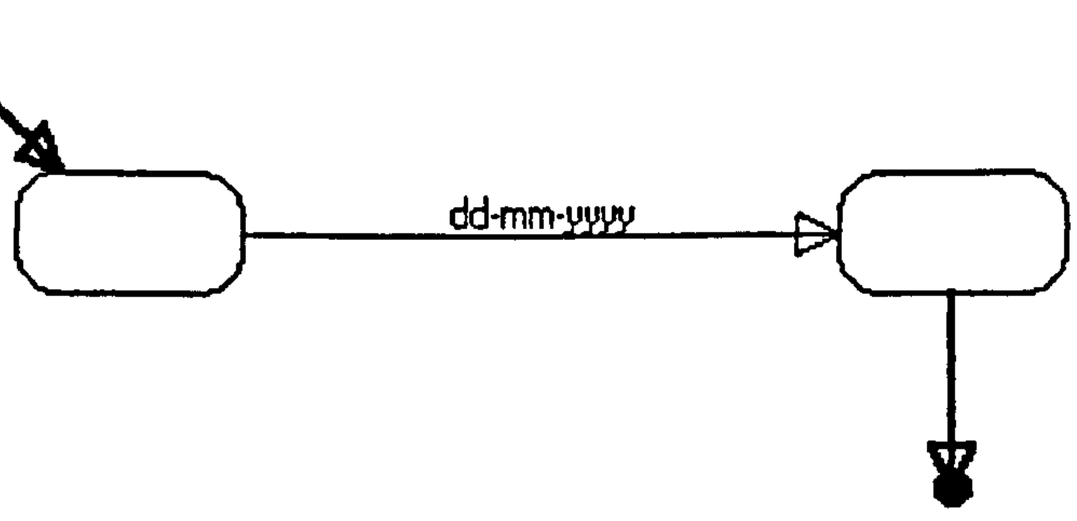


Diagram
Contact Number
(Con_no or Con_nr)



Diagram

Date

F10

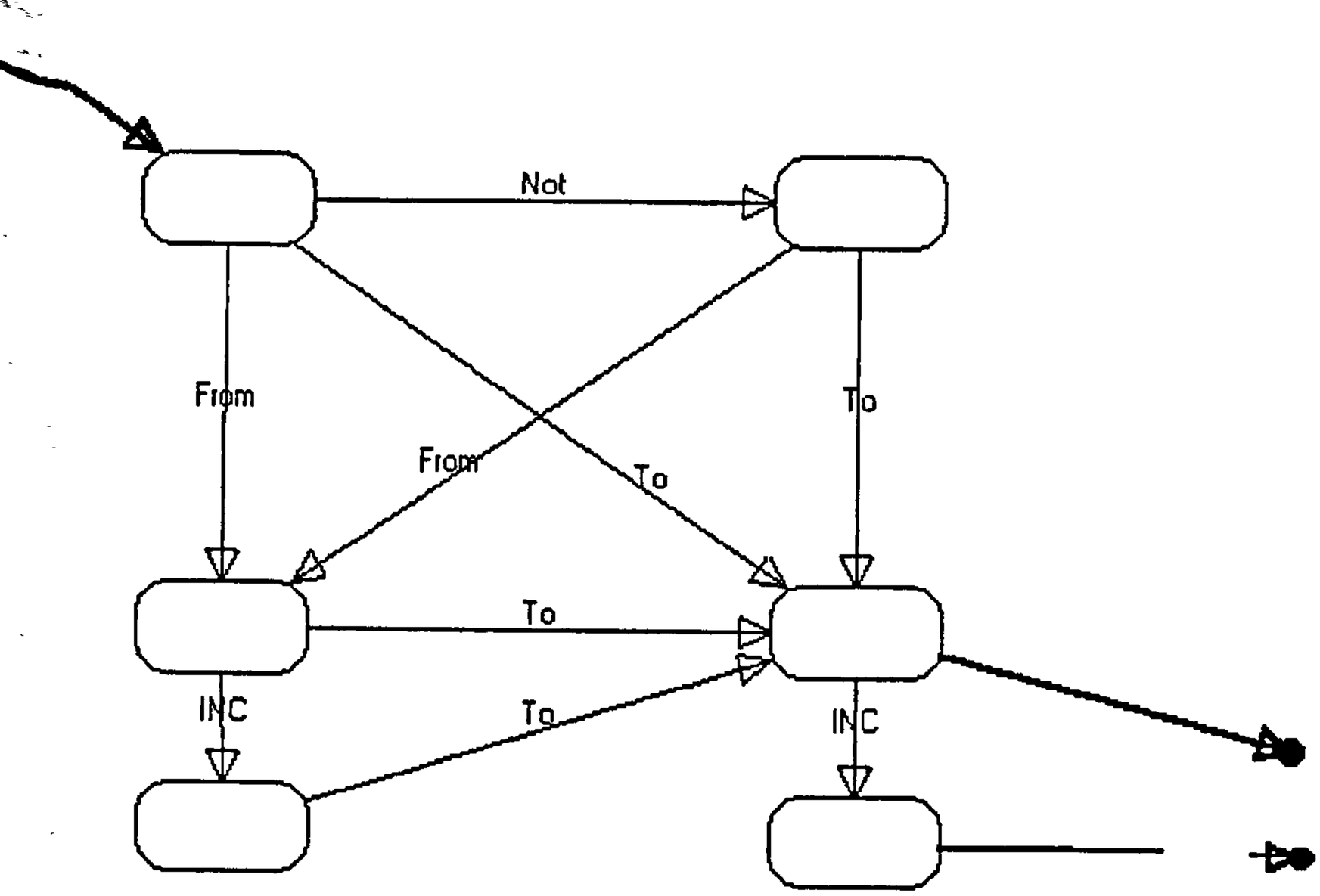


Diagram
Date Range
(DR)

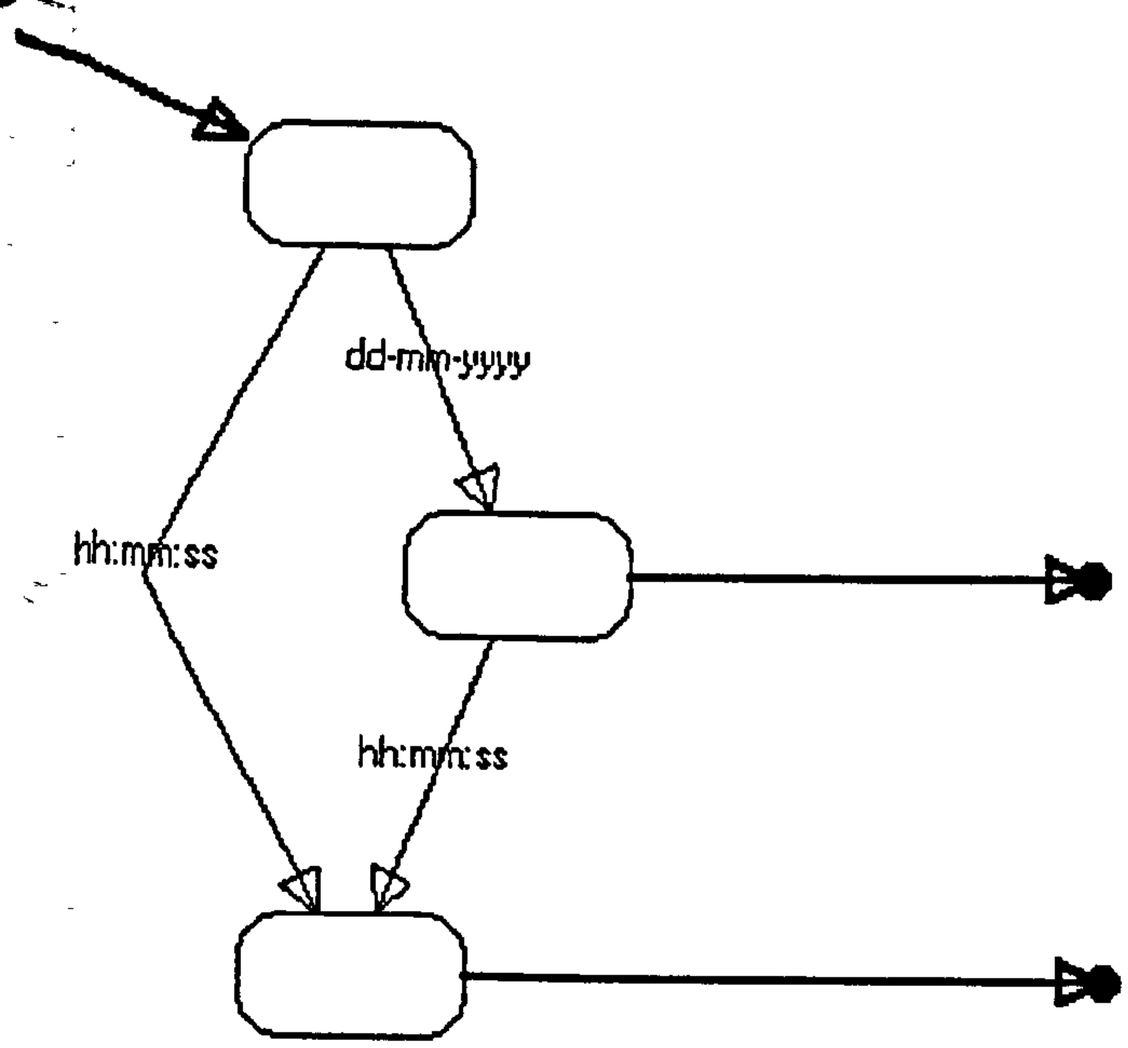


Diagram
Date and Time
(DT)

F12

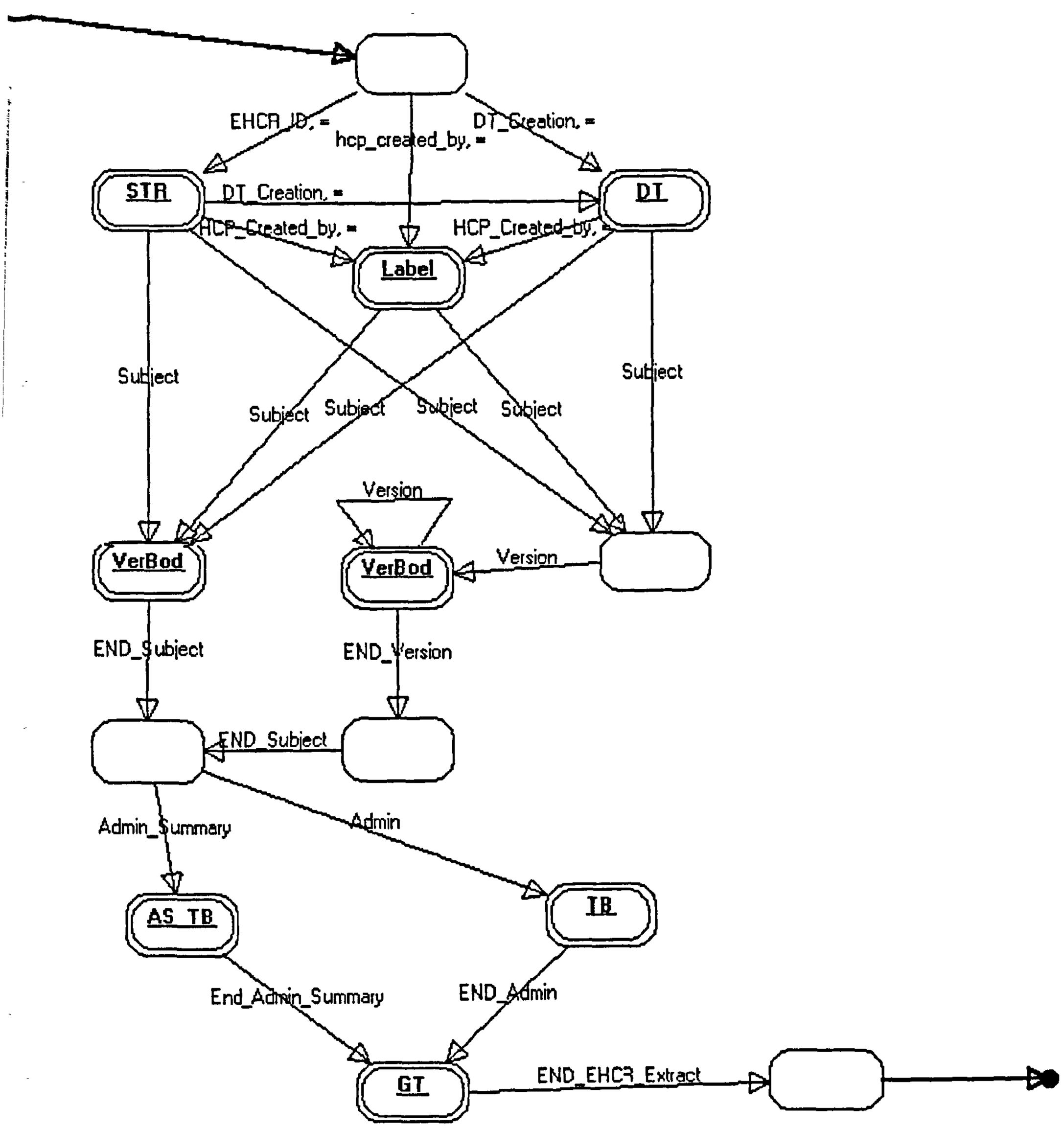
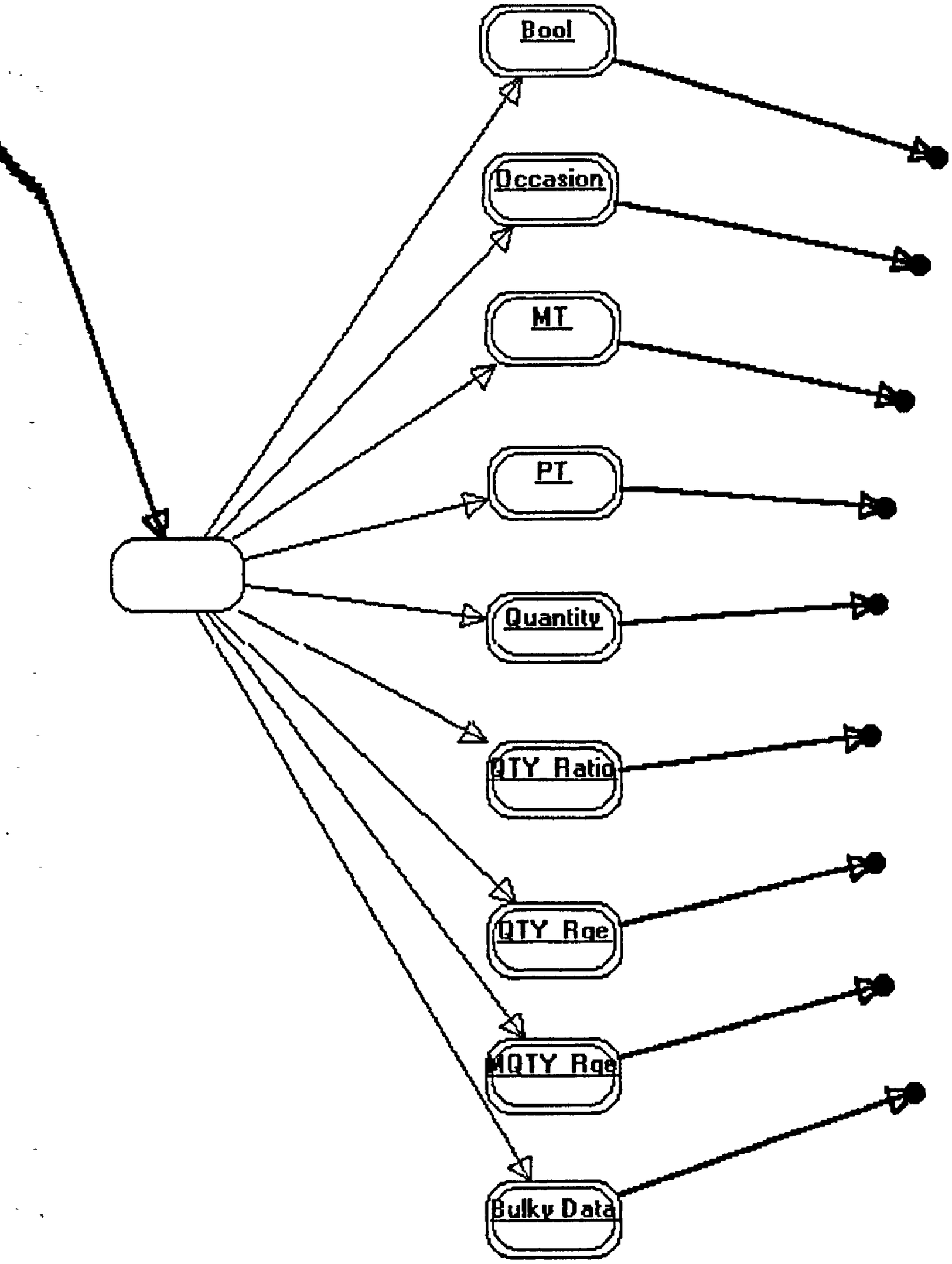


Diagram
EHCR Extract
(Extract)

F13



NB: Quantity, QTY_Ratio, QTY_Rge,and MQTY_Rge are not modelled in these diagrams

Diagram
EHCR Information
(E_Info)

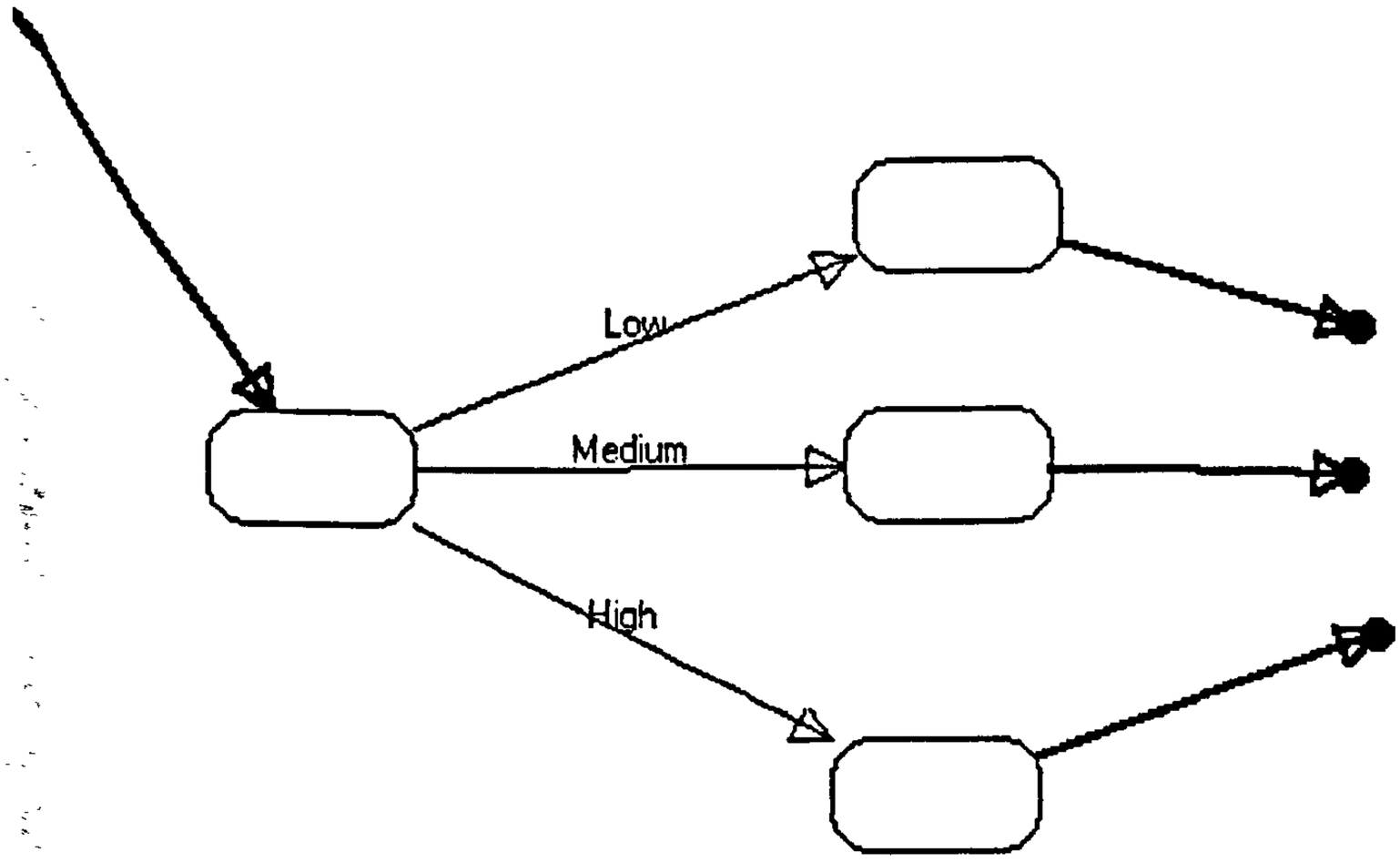


Diagram
Emphasis Level
(E_level)

F15

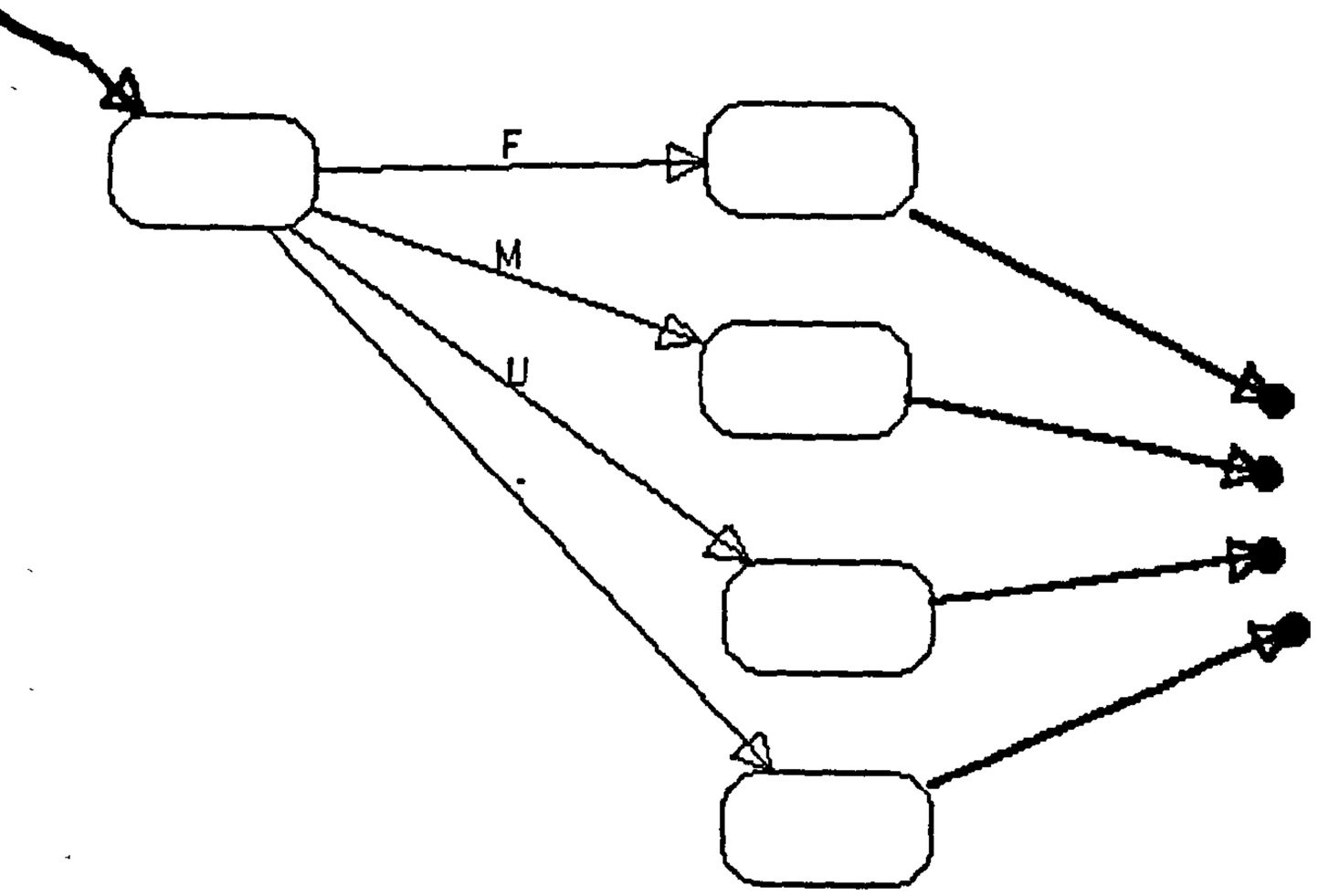


Diagram
Gender Code
(Gen Code)

F16

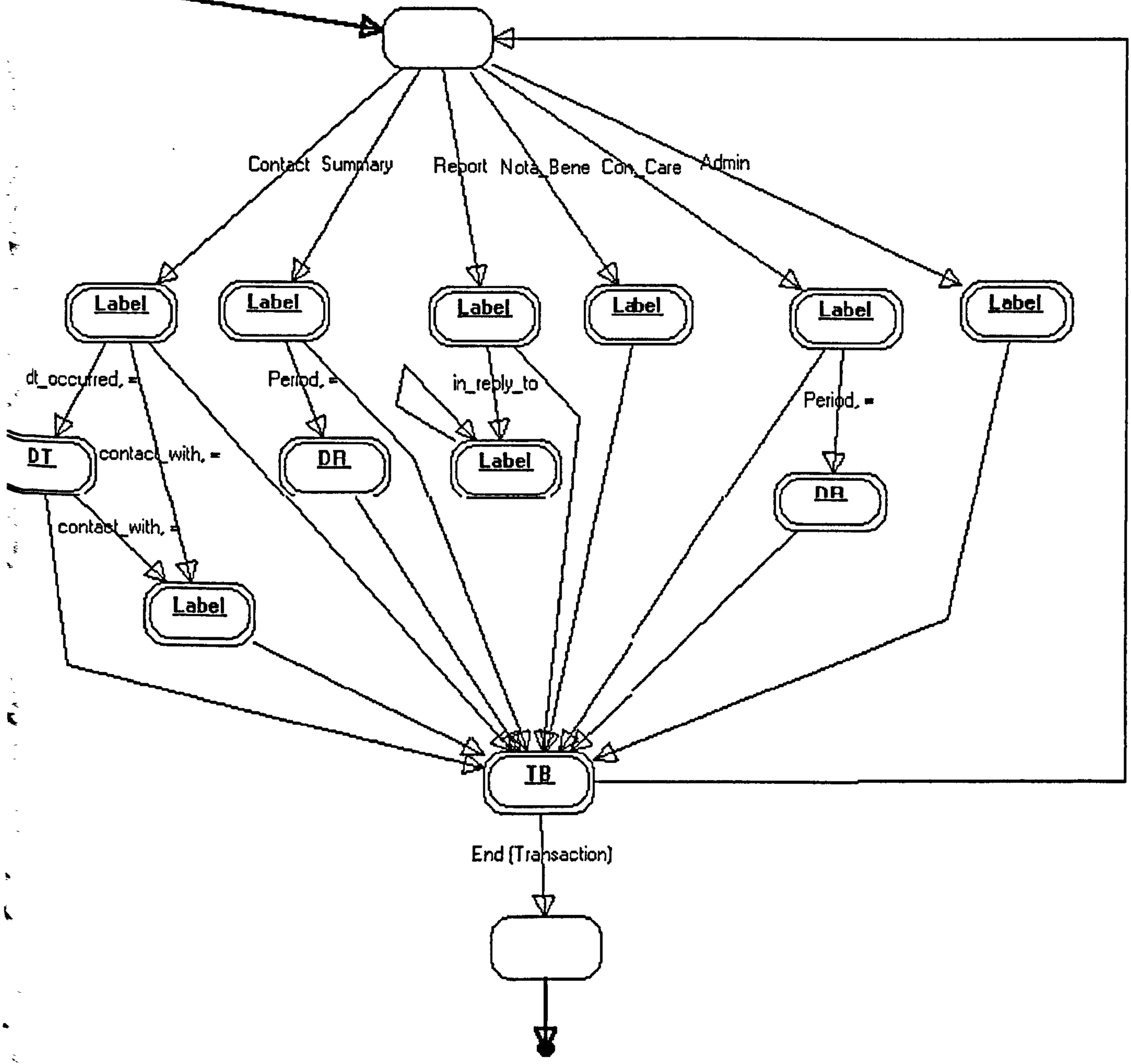


Diagram
General Transaction
(GT)

F17

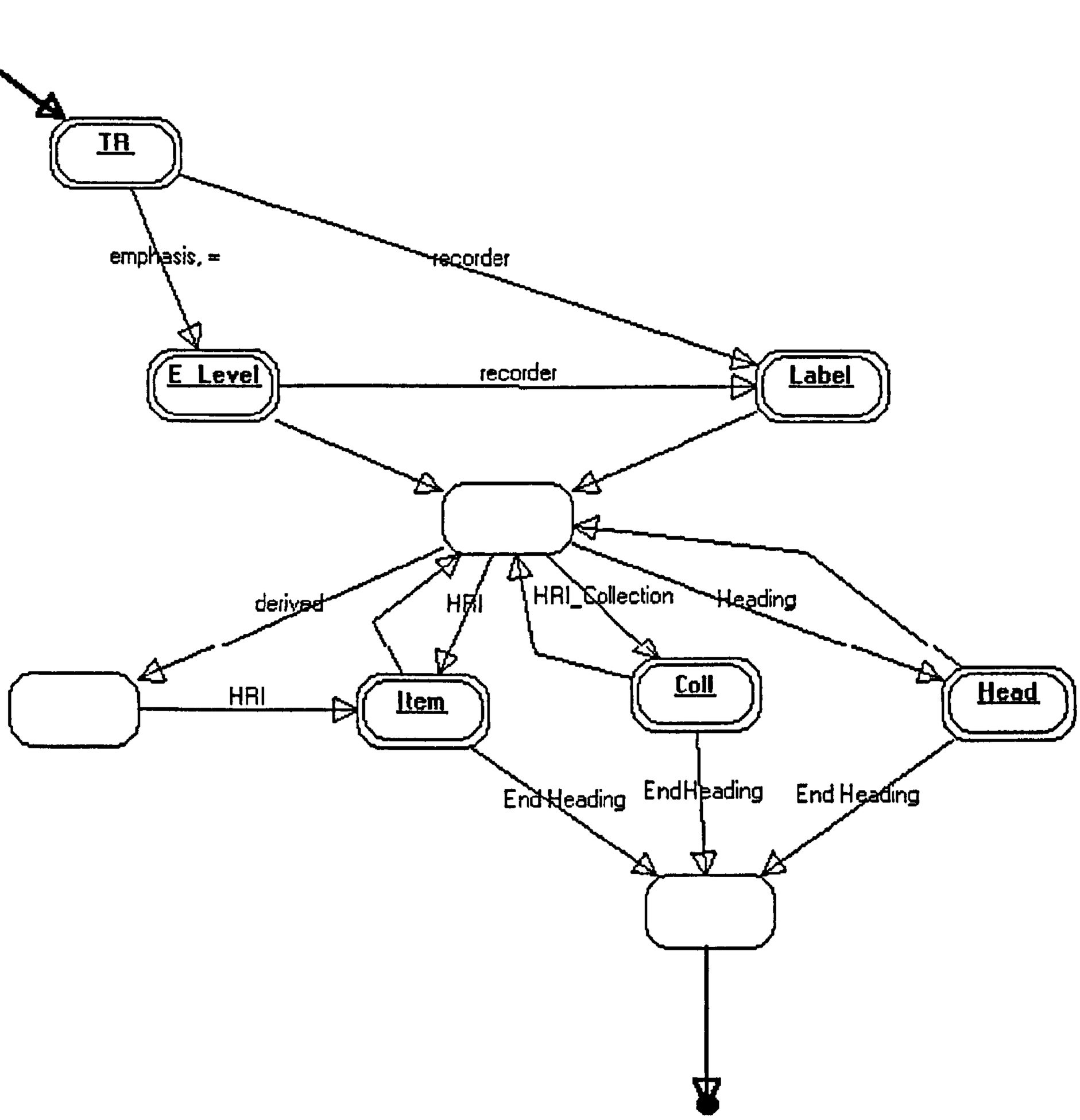


Diagram
Heading
(Head)

F18

TEXT BOUND INTO

THE SPINE

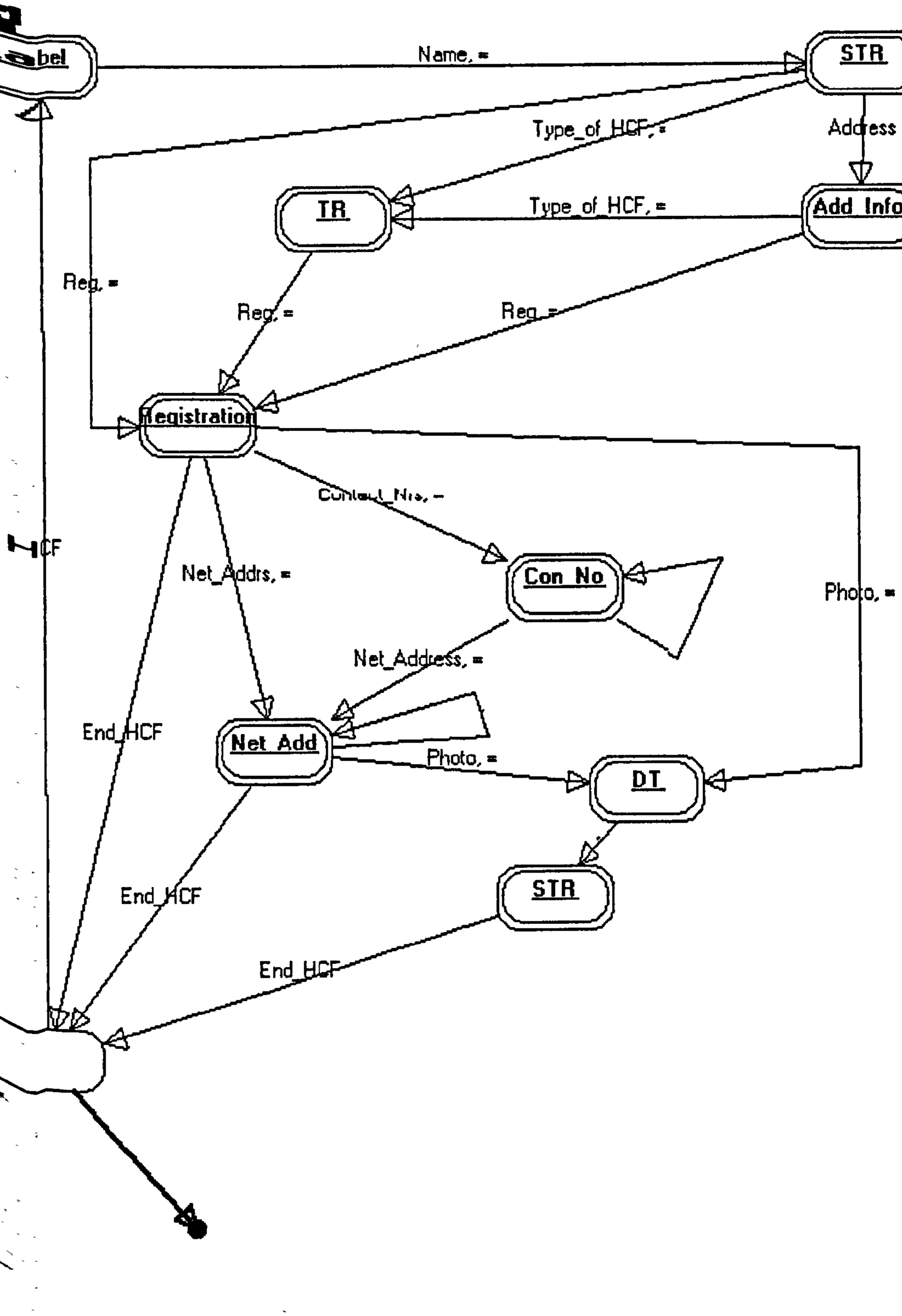
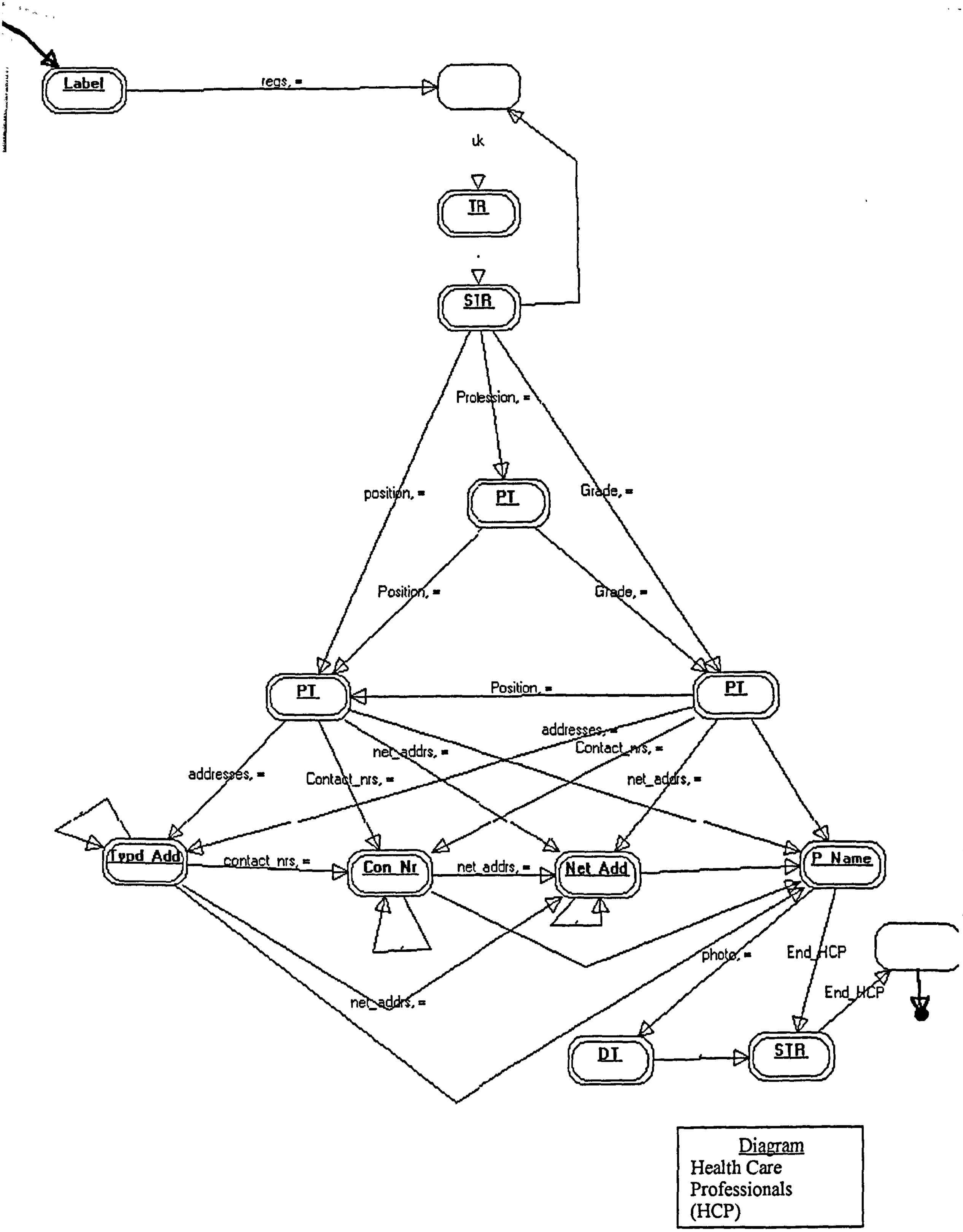


Diagram
Health Care
Facilities
(HCF)


Label
regs, =
uk
IR
SIR
PI
Profession, =
position, =
Grade, =
Position, =
Grade, =
PT
PT
Position, =
addresses, =
Contact_nrs, =
net_addrs, =
addresses, =
Contact_nrs, =
net_addrs, =
Typd Add
contact_nrs, =
Con Nr
net_addrs, =
Net Add
P Name
photo, =
End_HCP
End_HCP
net_addrs, =
DI
SIR
Diagram
Health Care
Professionals
(HCP)

F20

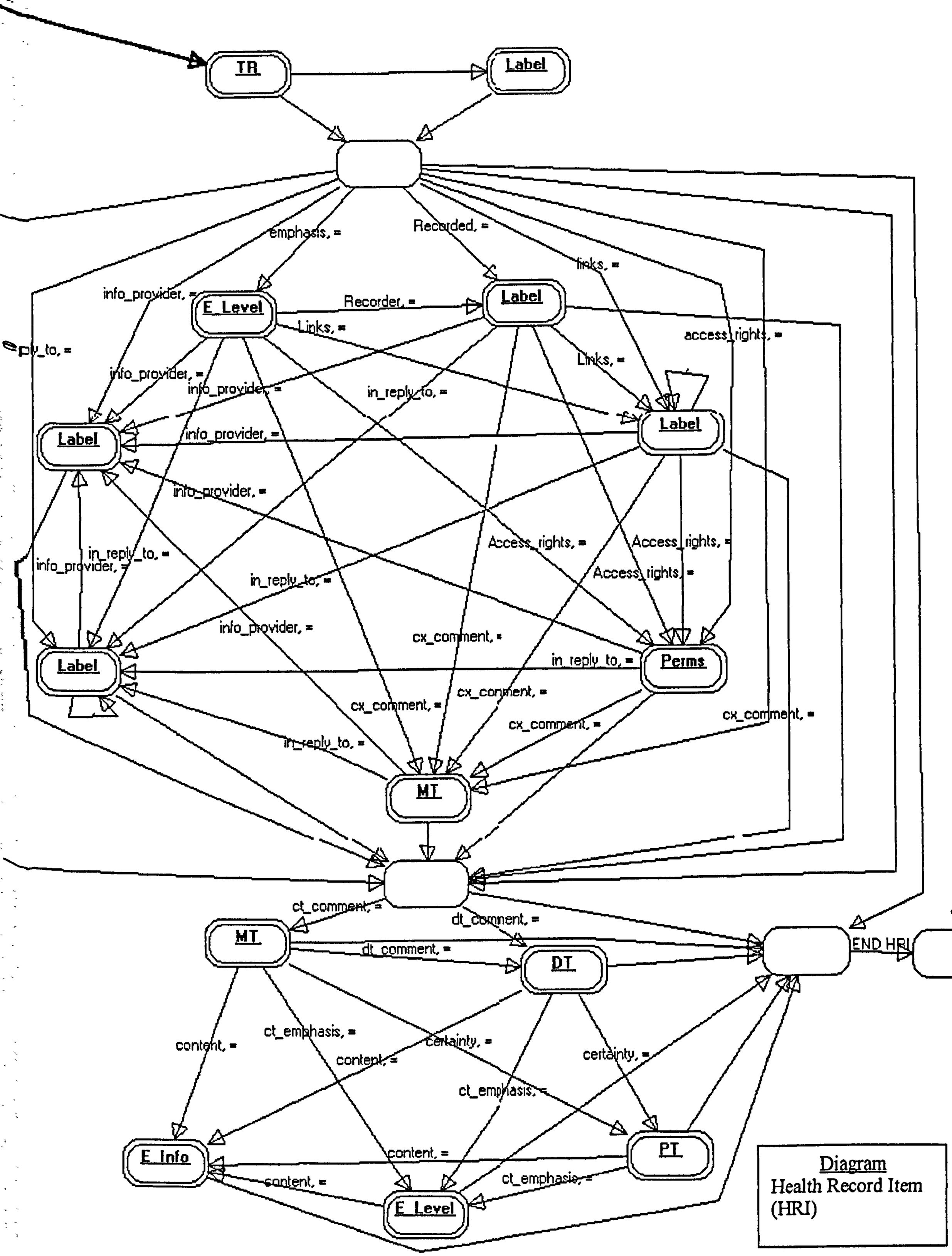
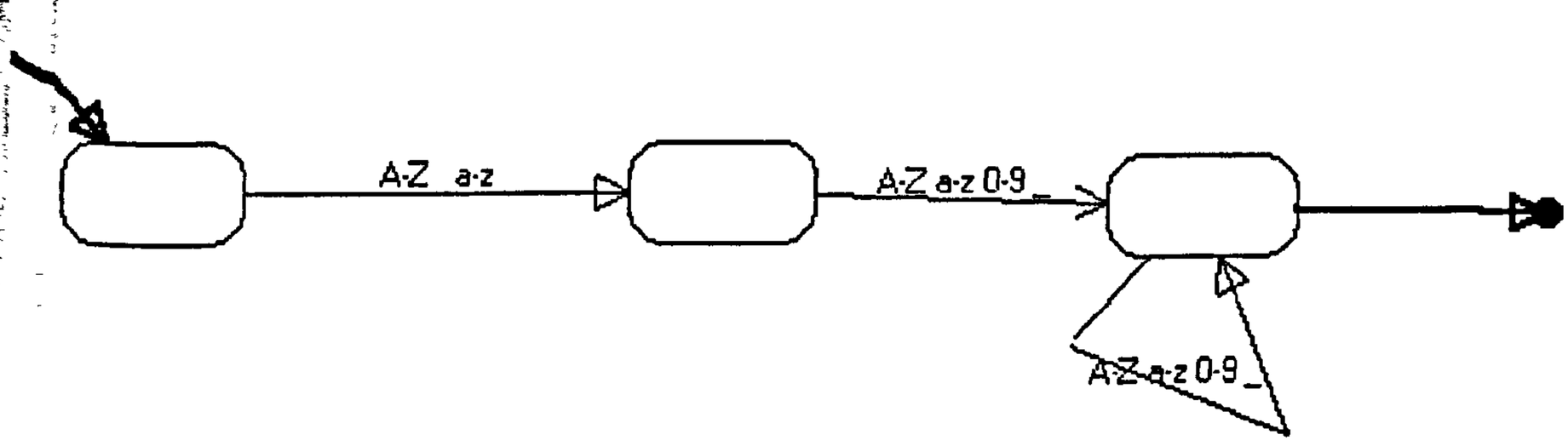


Diagram
Health Record Item
(HRI)

F21



Diagram

Label

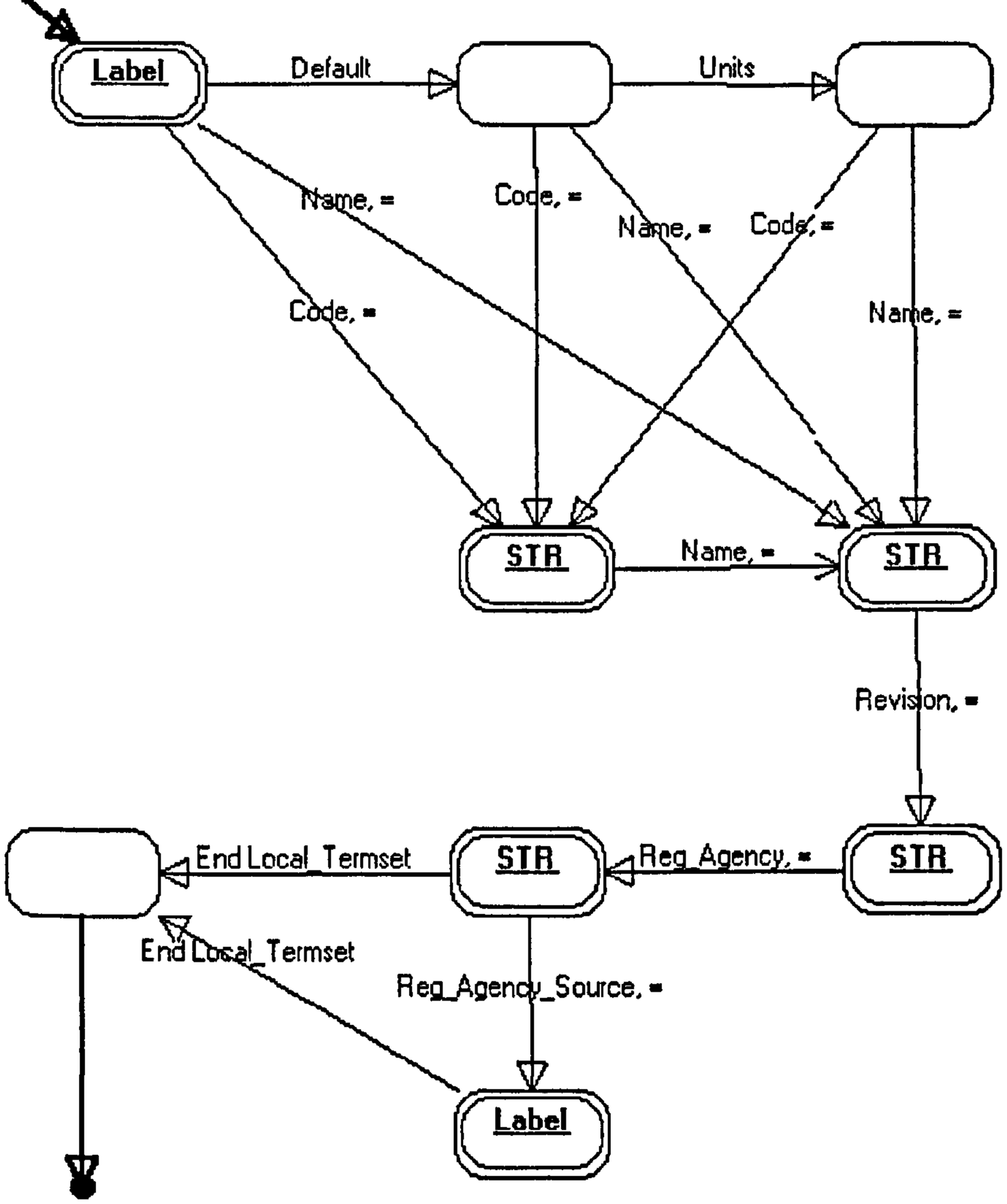


Diagram
Local Termset
(L_Termset)

F23

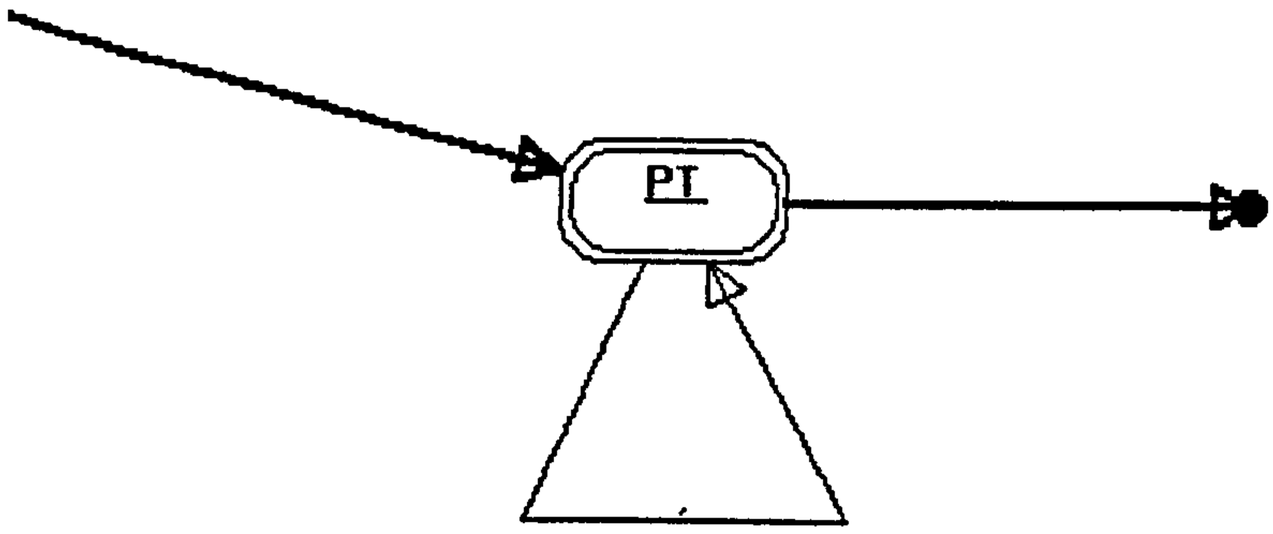


Diagram
Multi Text
(MT)

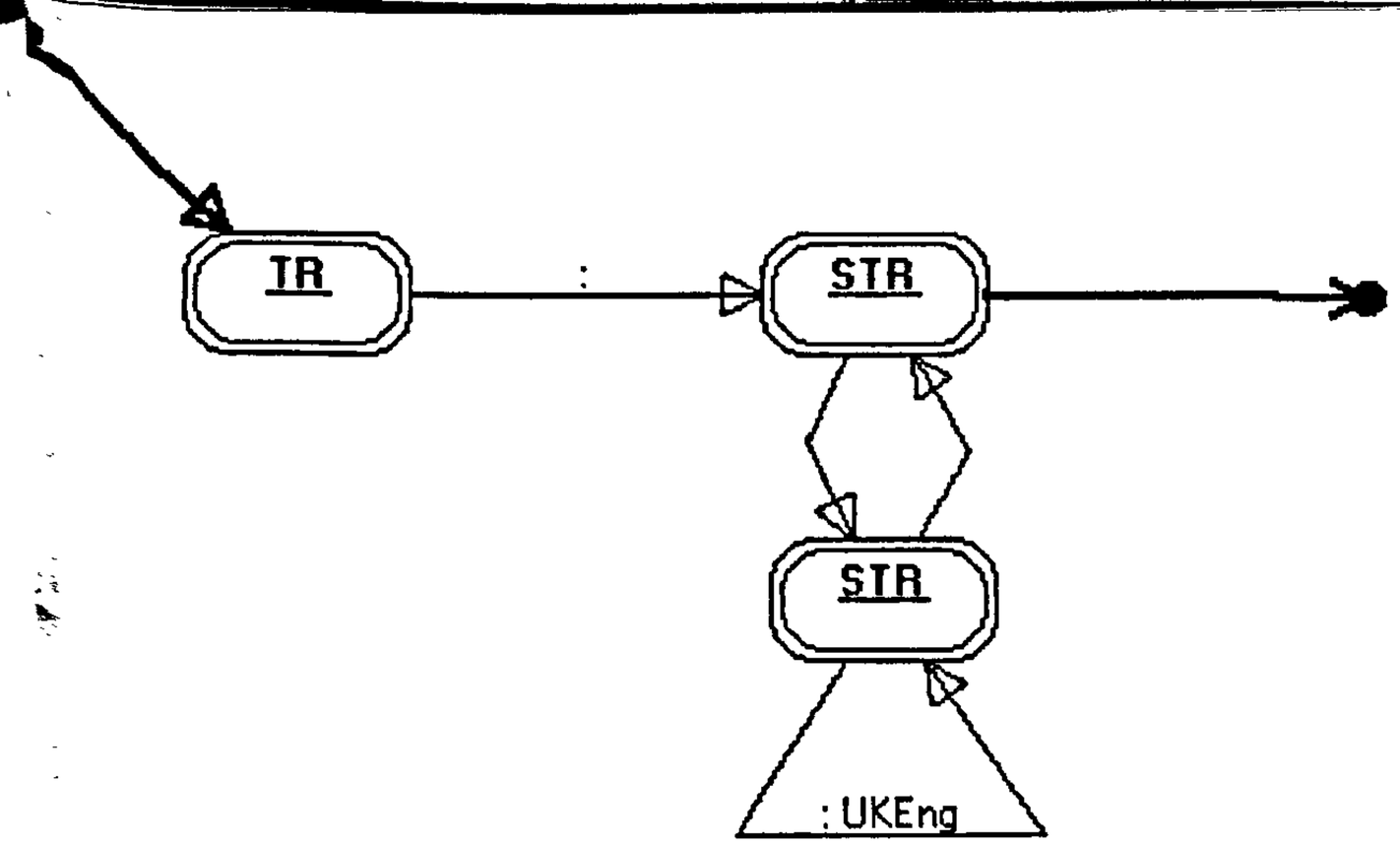


Diagram
Net Address
(Net_Add)

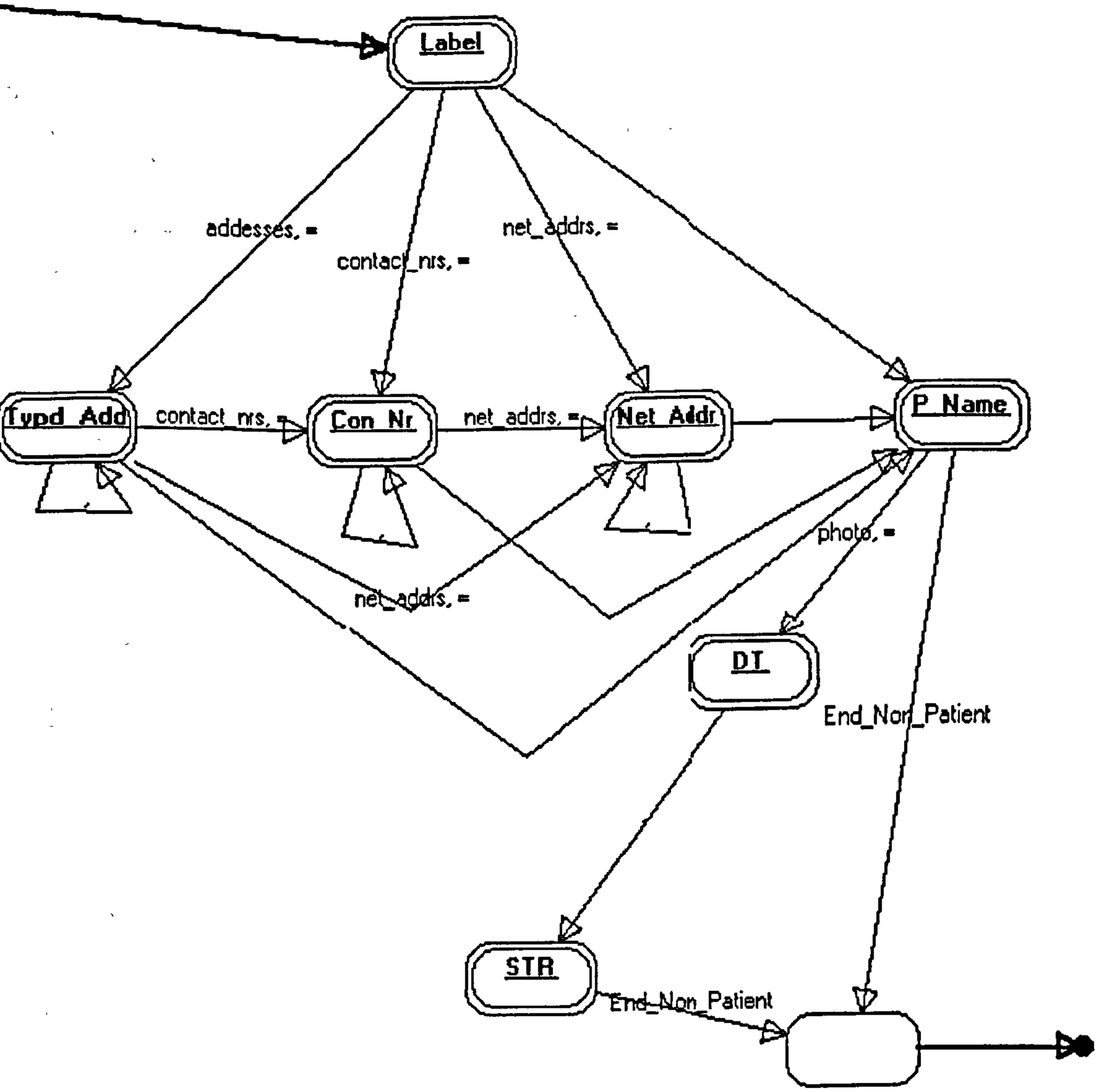


Diagram
Non-Patient
(Non-Pat)

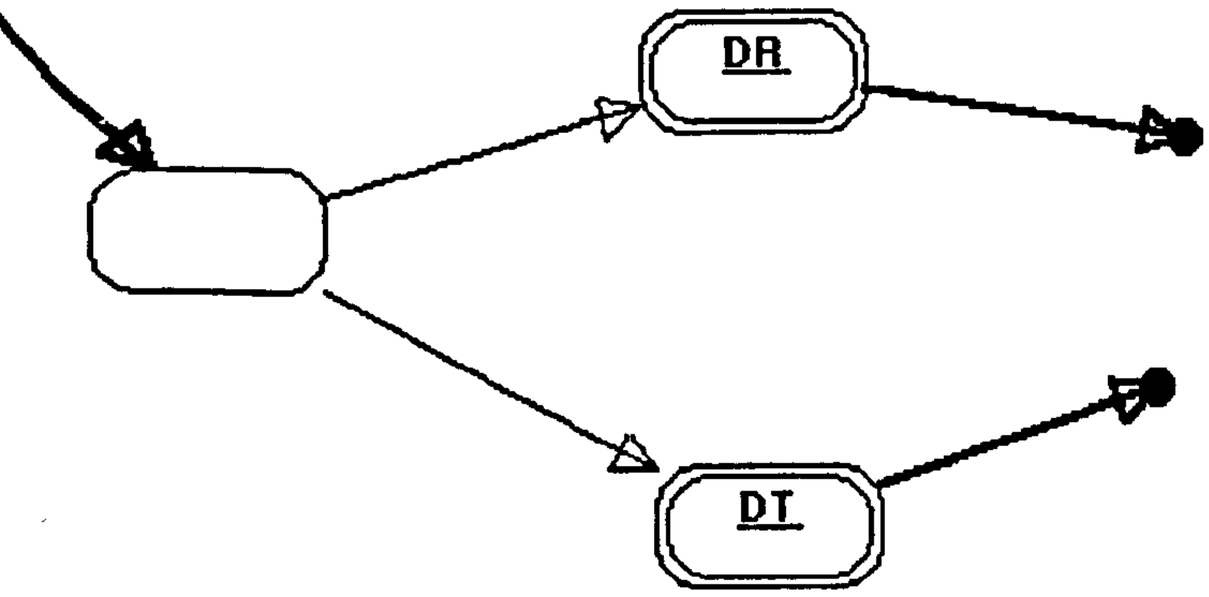


Diagram
Occasion

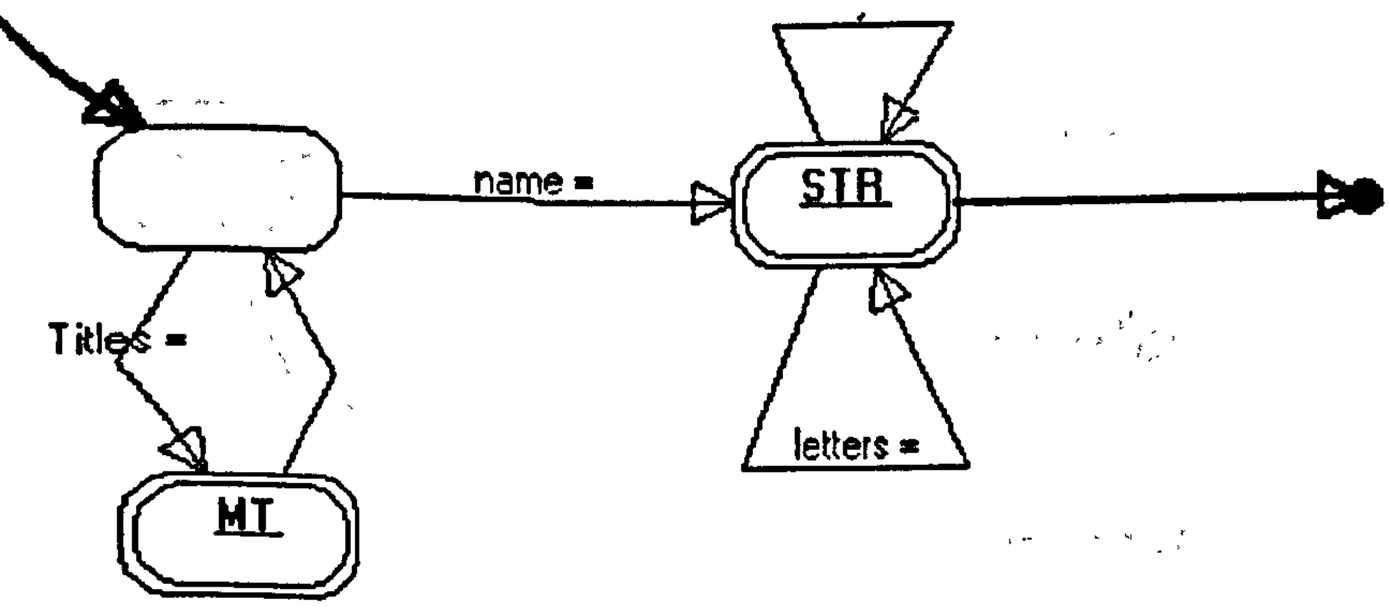


Diagram
Patient Name
(P_Name)

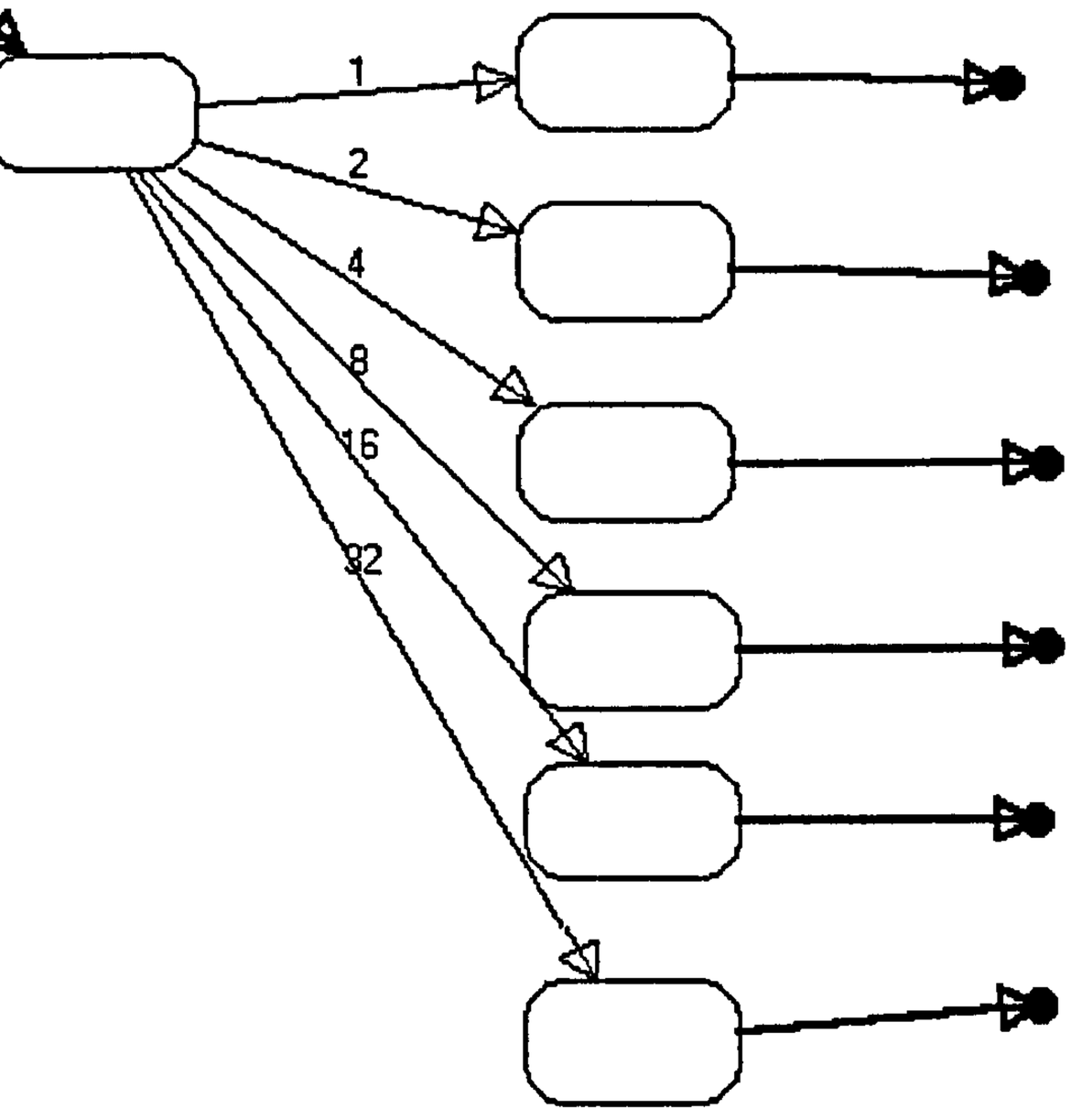


Diagram
Perms

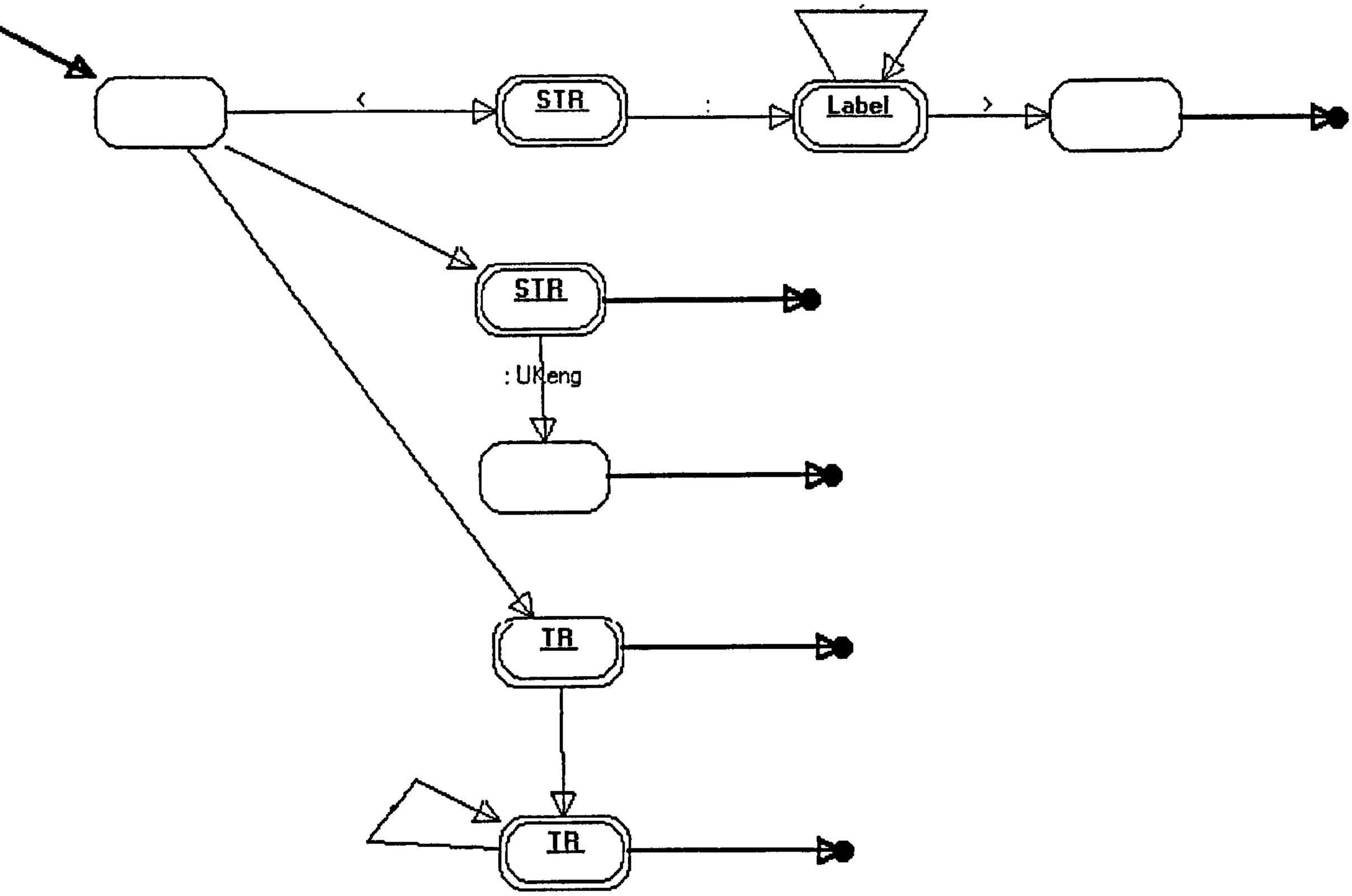


Diagram
Plain Text
(PT)

F30

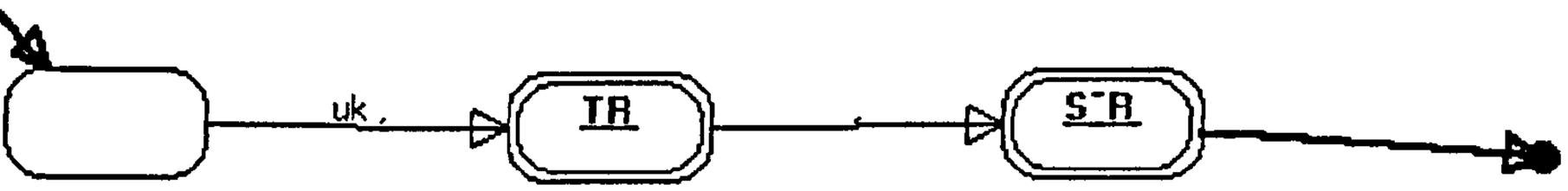


Diagram
Registration

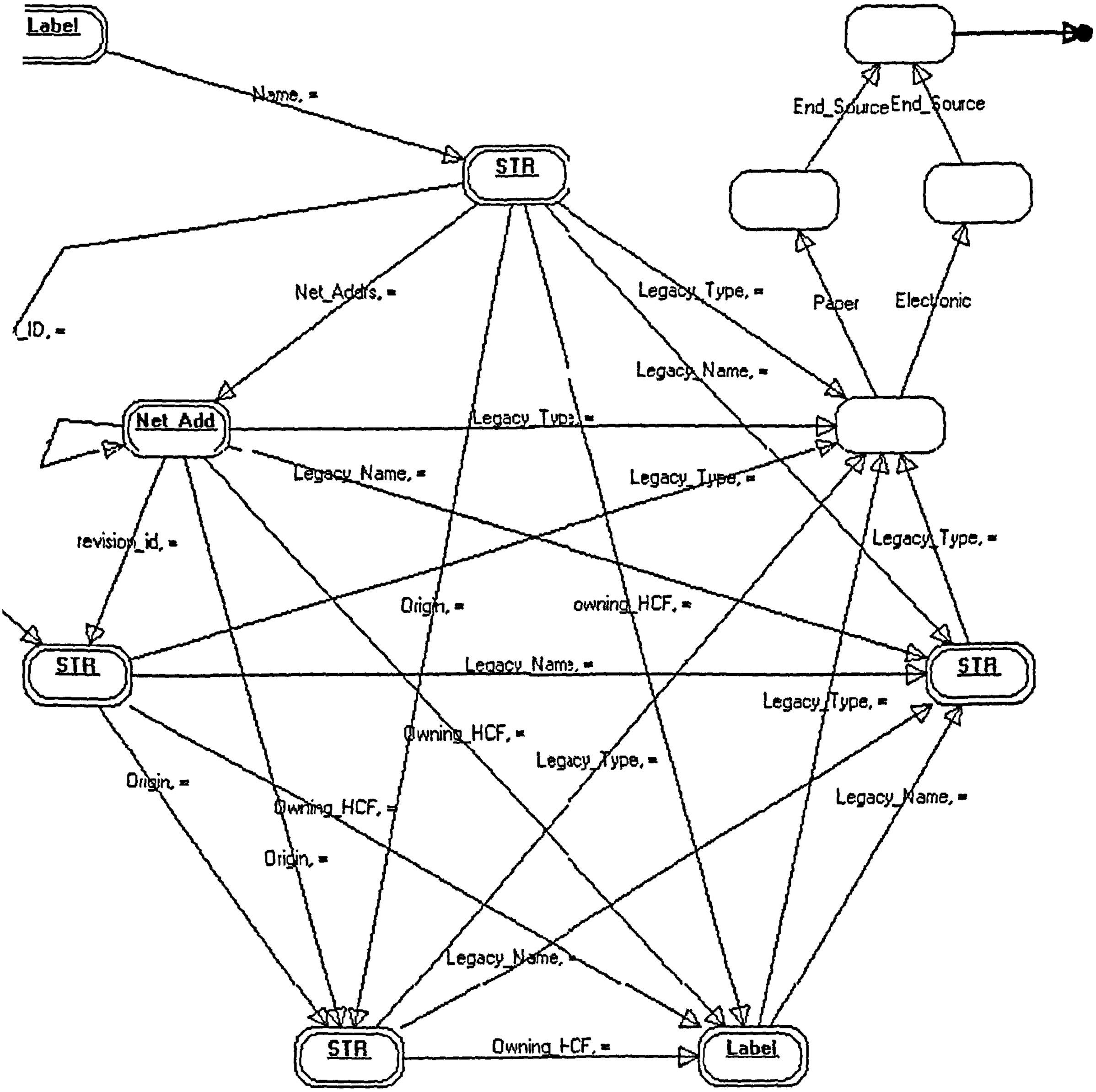


Diagram
Source

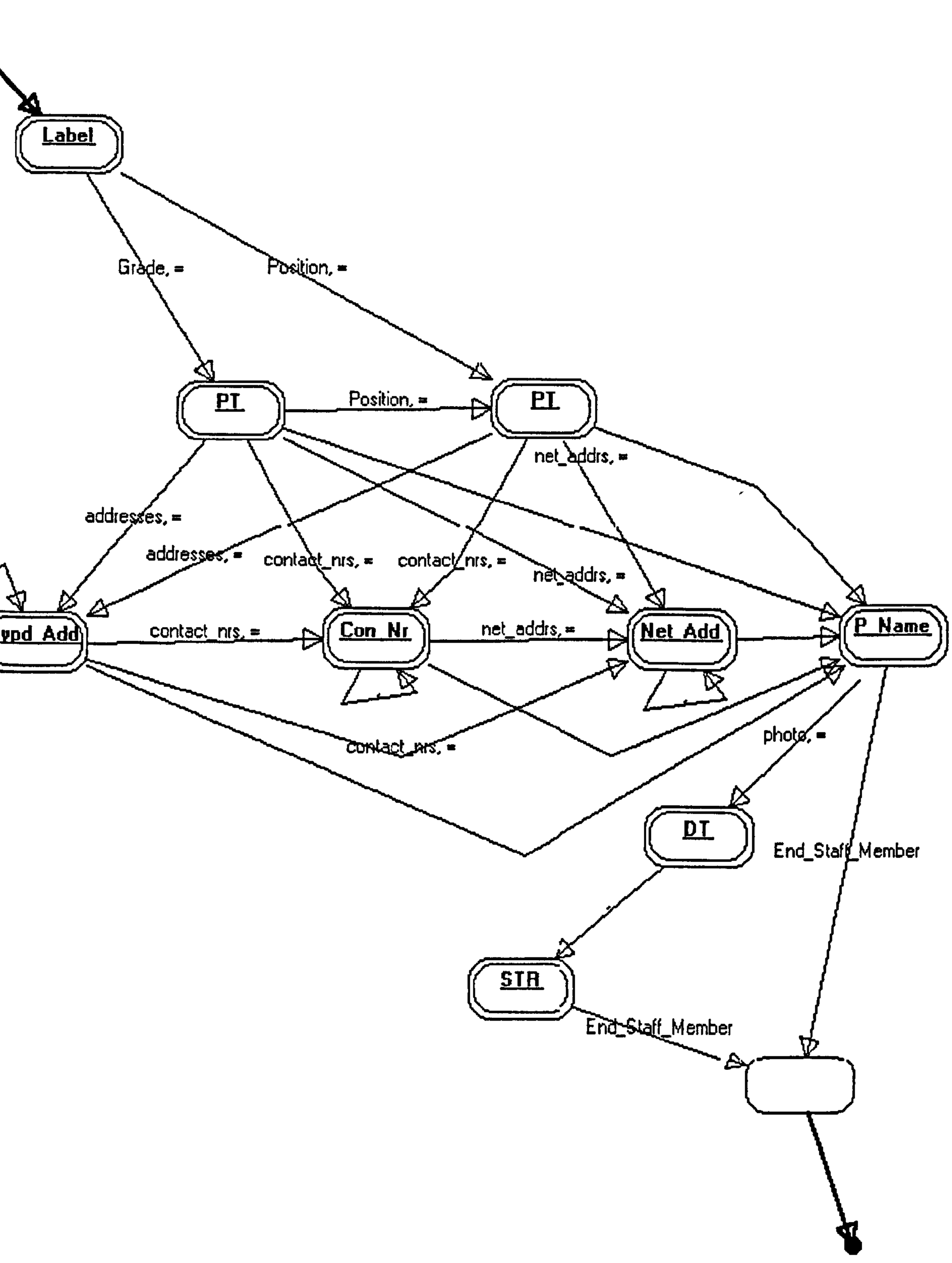


Diagram
Staff Member
(Staff_M)

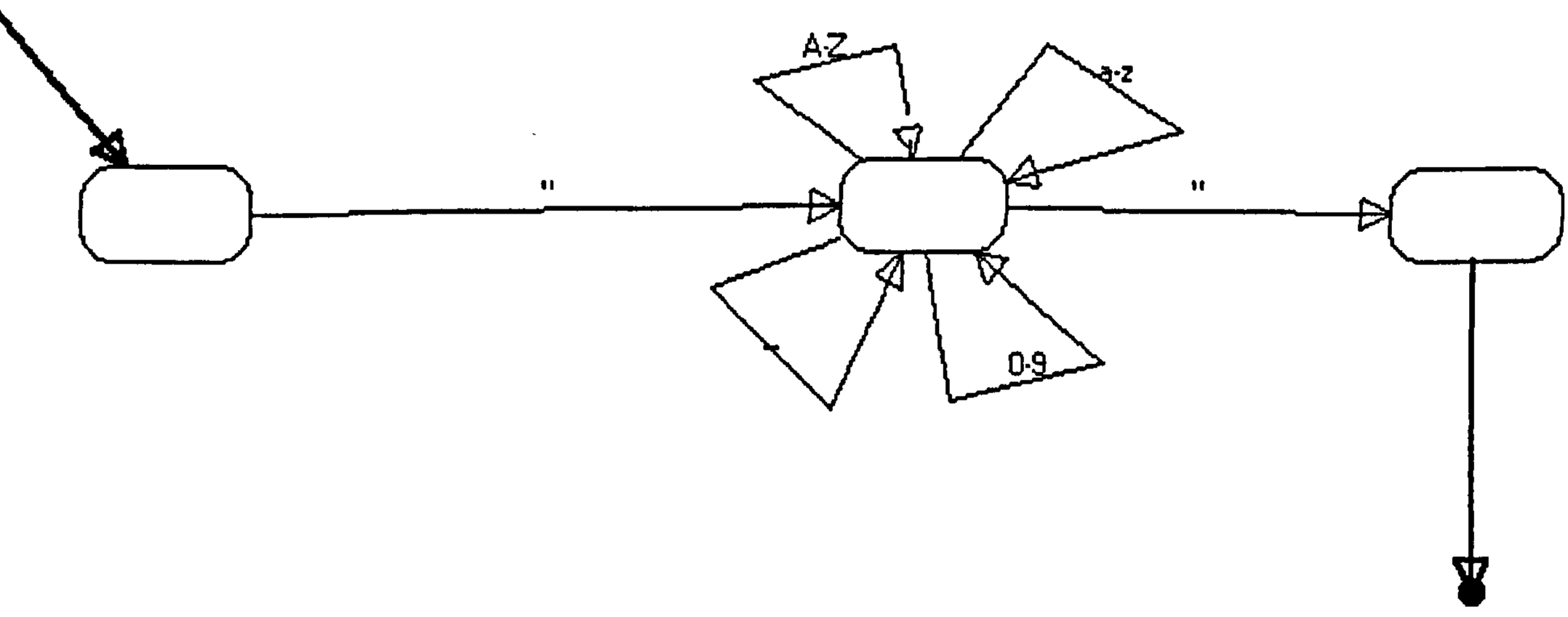


Diagram
String
(Str)

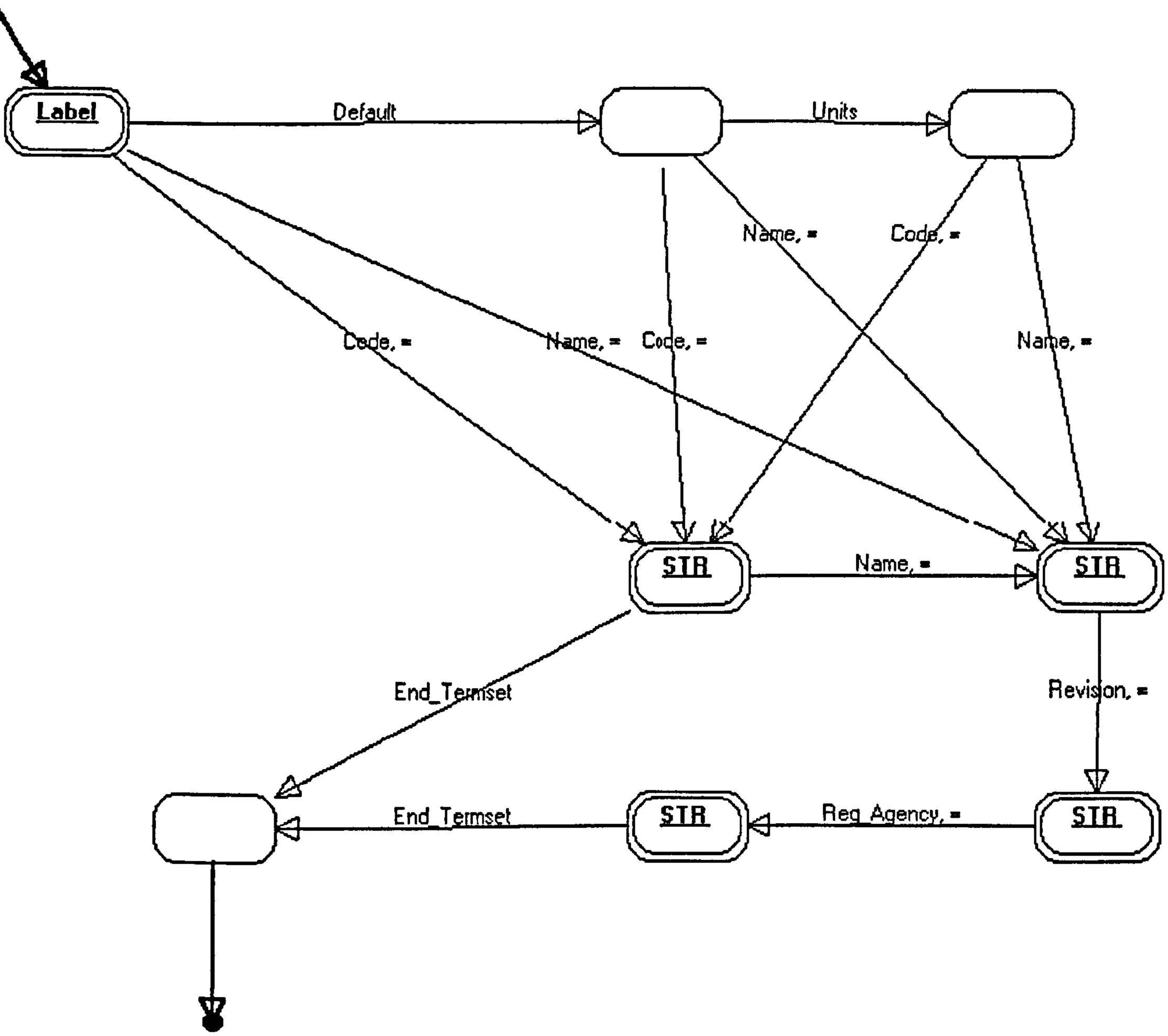


Diagram
Termset

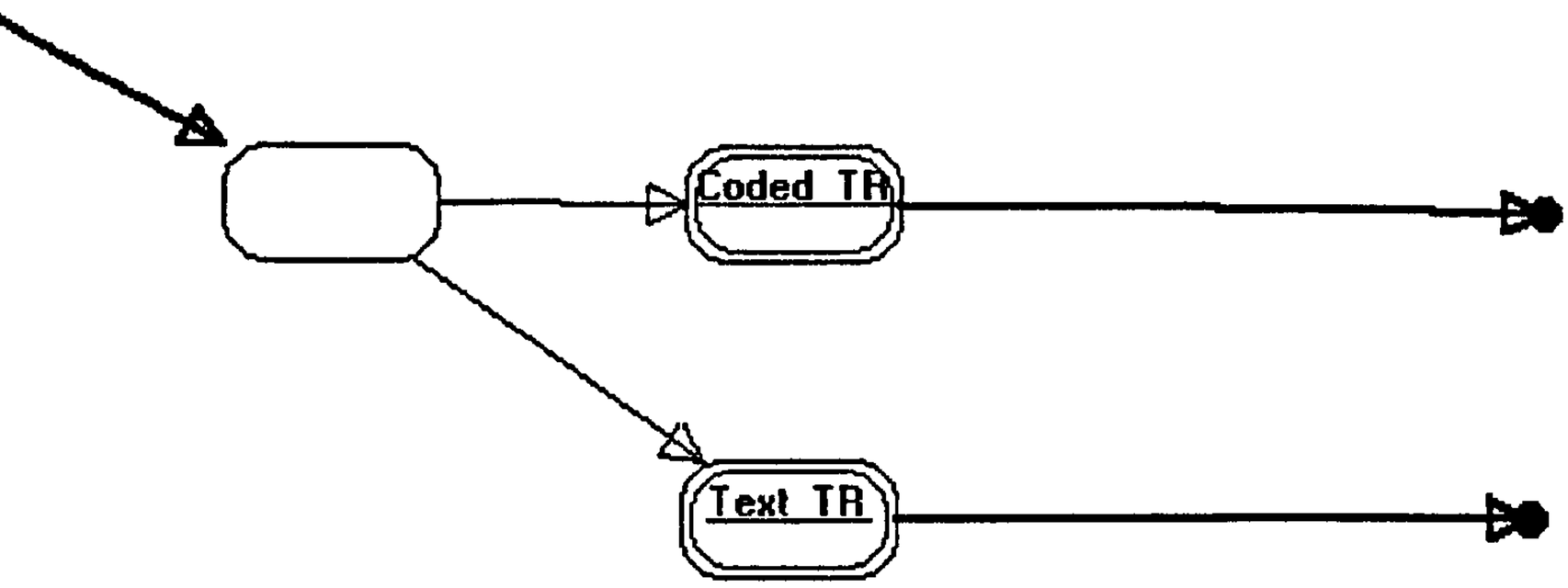


Diagram
Term Reference
(TR)

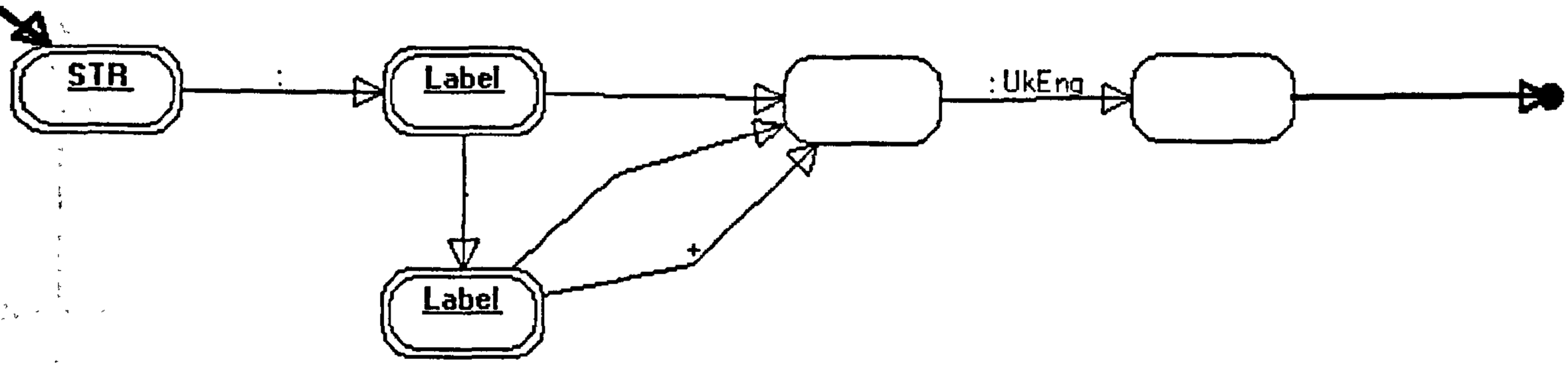


Diagram
Text Term
Reference
(Text_TR)

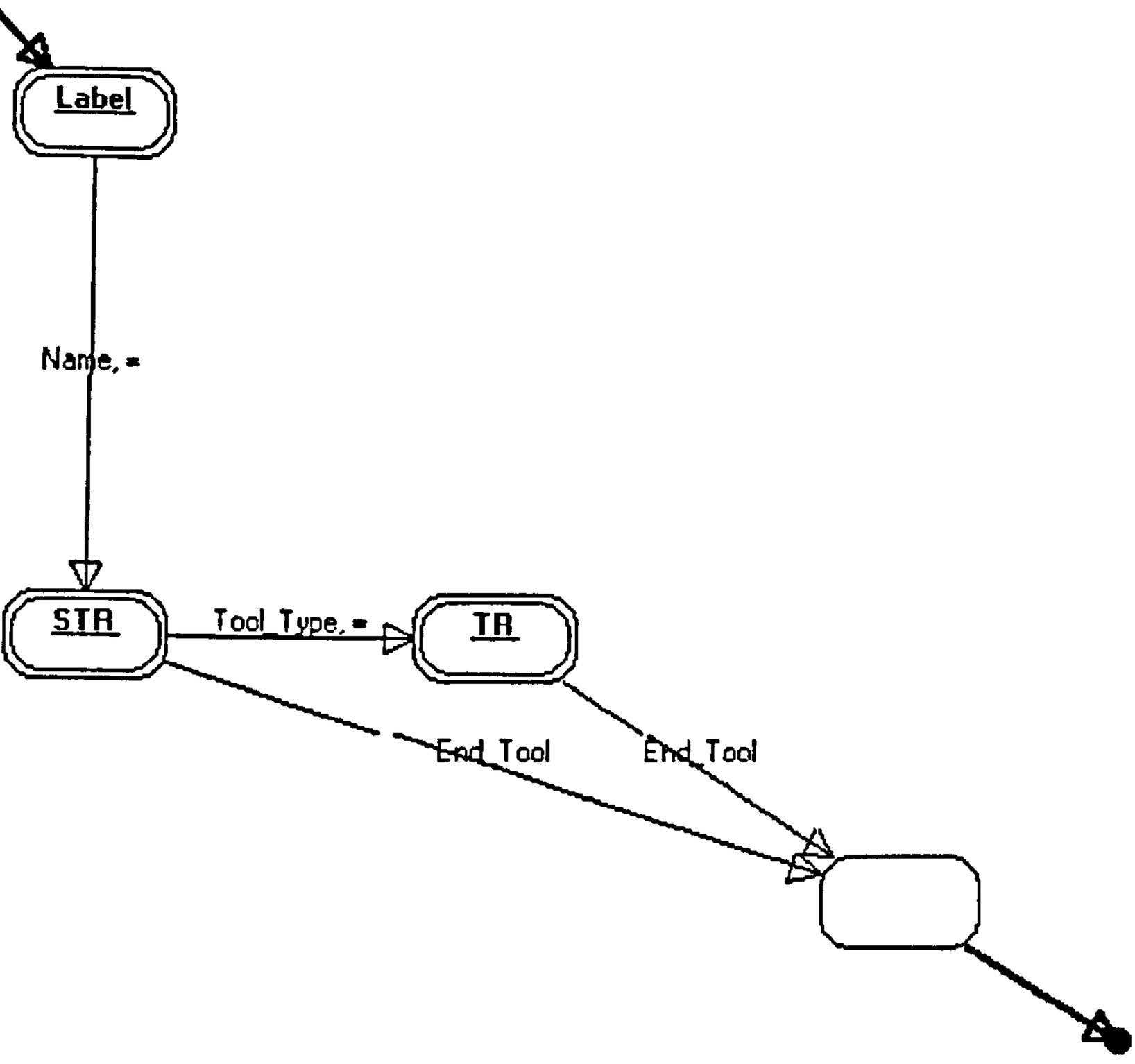

Label
Name, =
STR
Tool Type, =
TR
End Tool
End Tool

Diagram
Tool

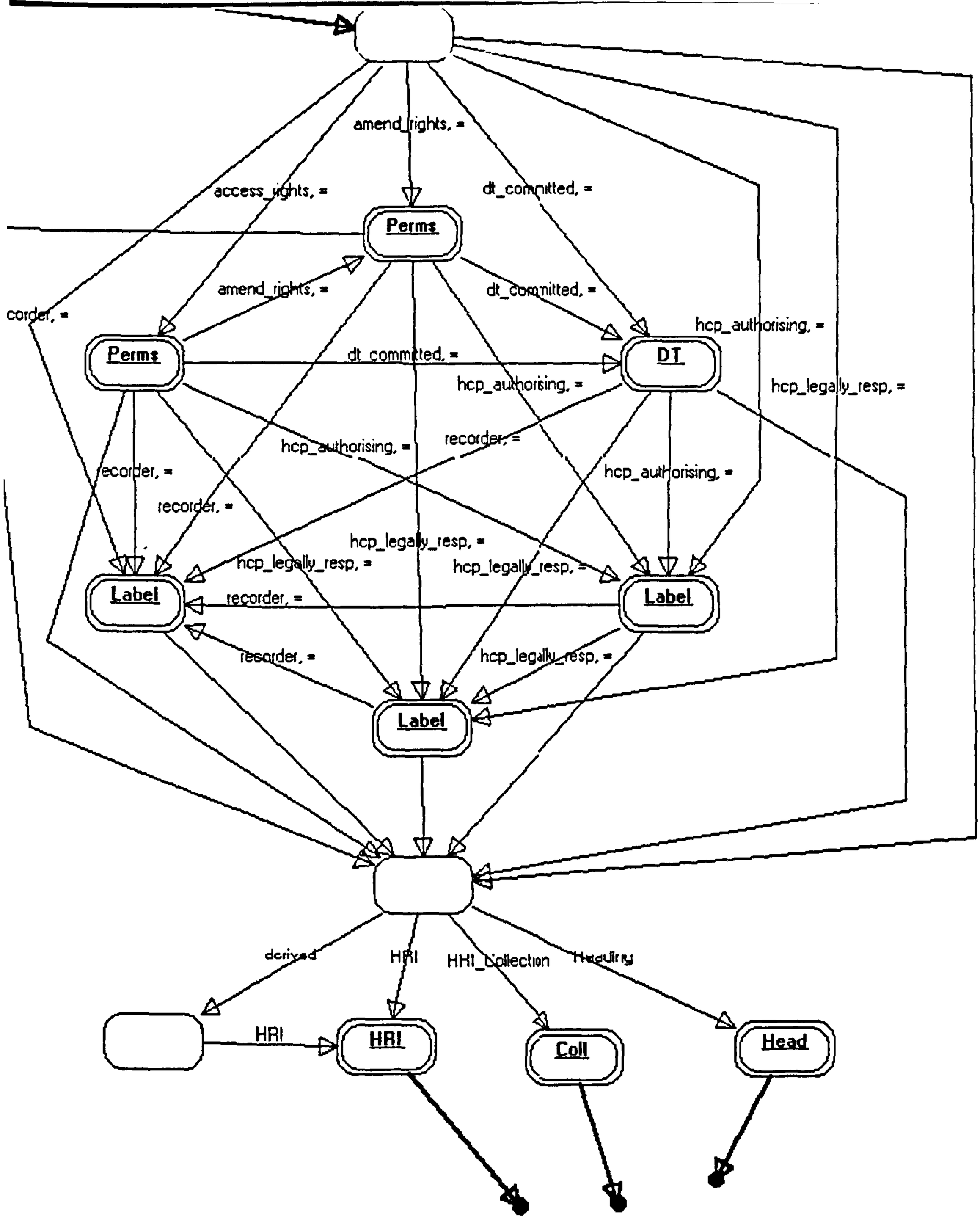


Diagram
Transaction Body
(TB)

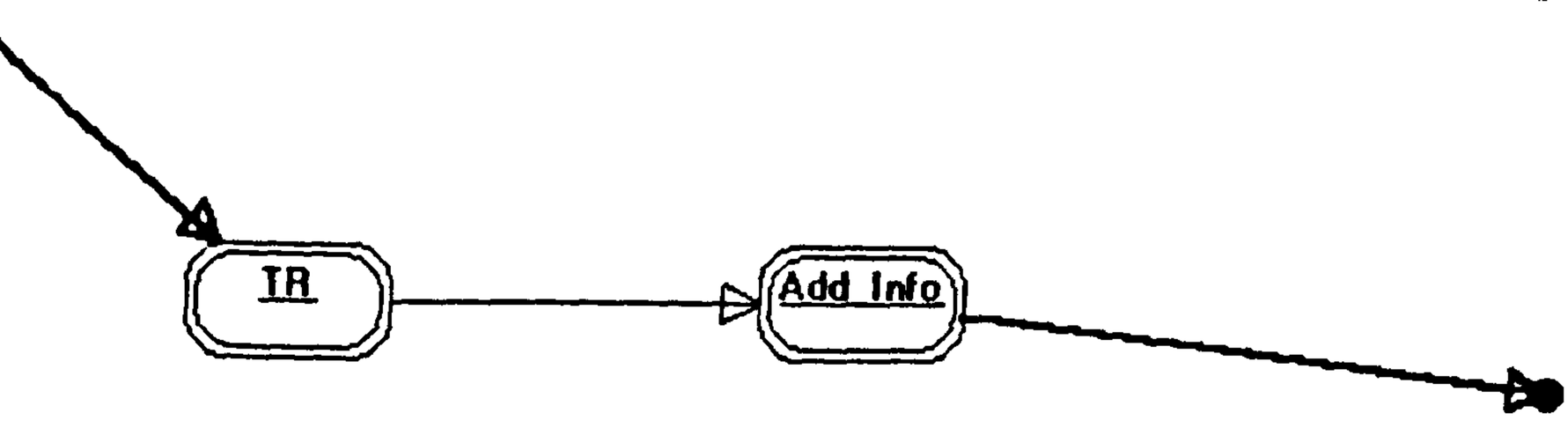


Diagram
Typed Address
(Typd_Add)

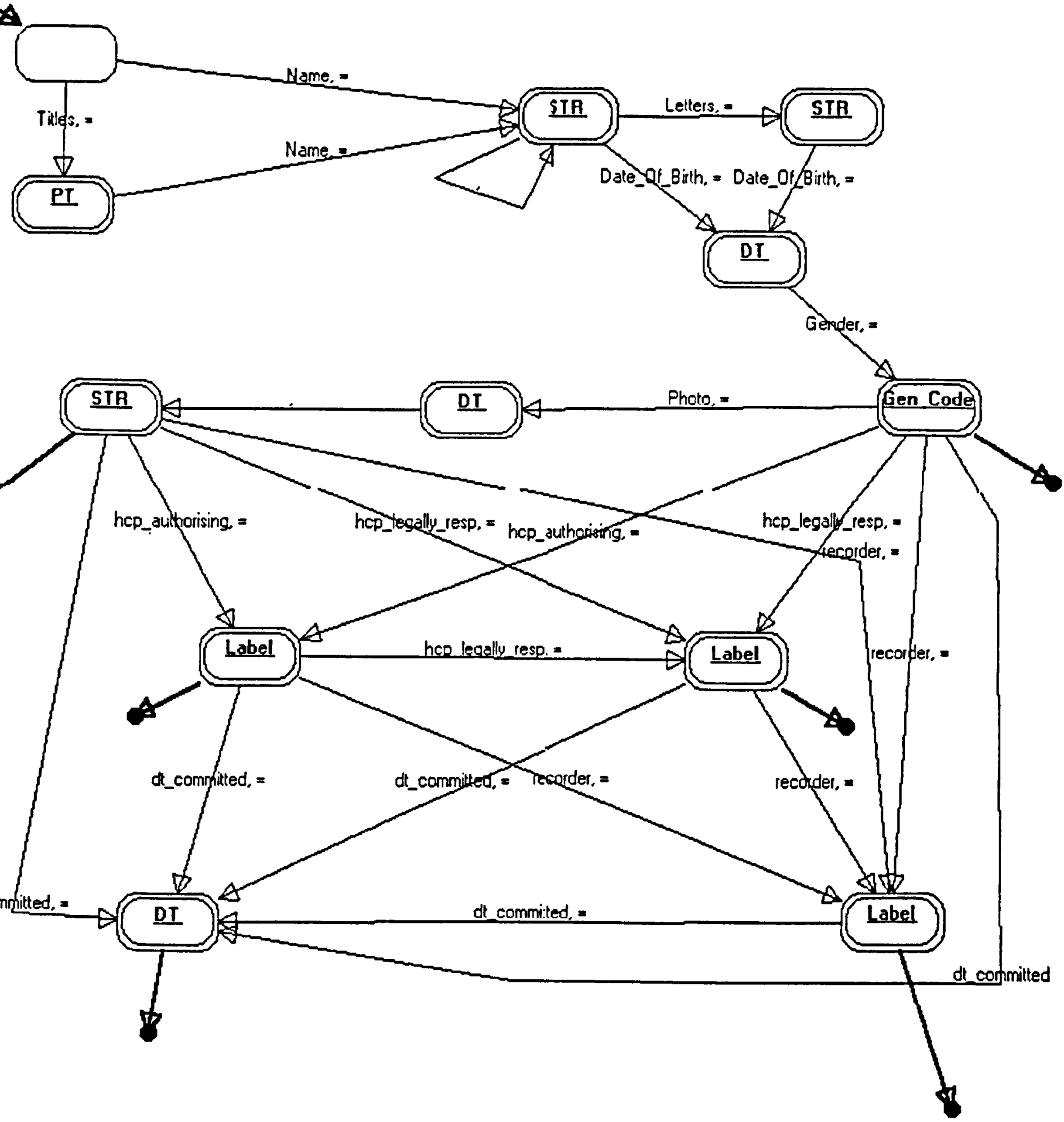


Diagram
Version Body
(VerBod)

Appendix G - CLIF Compiler Output

This appendix shows example data that has been generated by the CLIF compiler. It shows legacy data in GEF for data from the Miniclinic database. For a further explanation of the CLIF compiler see section 6.3.

```
(P,C,83) GEF
  SEQUENCE
    [0] SET
      (P,C,25) TERMSET
        SET
          [0] OCTETSTRING = "0123"
          [1] OCTETSTRING = "miniclinicTS1"
          [2] OCTETSTRING = "1.0"
          [4] (P,C,26) REG-AGENCY
            SET
              [1] OCTETSTRING = "Termset agency"
[2] SET
  (P,C,69) HCP
    SET
      (P,C,78) REGISTRATION
        SET
        [0] OCTETSTRING = "uk"
        [1] (P,C,24) TERM-REF
          SET
            (P,C,22) PLAIN-TEXT
              SET
                (P,C,21) EHCR-INFO
                  SET
                    OCTETSTRING = "test"
                    [0] OCTETSTRING = "GarysReg"
          [2] OCTETSTRING = "STR Gary's registration"
        (P,C,68) STAFF-MEMBER
          SET
            (P,C,67) NON-PATIENT
              SET
                (P,C,66) PERSON
                  SET
                    (P,C,65) PROVIDER
                      INTEGER = "1"
                    [0] (P,C,73) PERSON-NAME
                      SET
                        [0] OCTETSTRING = "Gary"
                        [1] OCTETSTRING = "Evans"
[6] SET
  (P,C,75) HCF
    SET
      [0] INTEGER = "1"
      [1] OCTETSTRING = "Manor Park"
      [4] (P,C,78) REGISTRATION
        SET
        [0] OCTETSTRING = "uk"
        [1] (P,C,24) TERM-REF
          SET
            (P,C,22) PLAIN-TEXT
              SET
```

```
            (P,C,21) EHCR-INFO
              SET
                OCTETSTRING = "test"
              [0] OCTETSTRING = "ManorParkReg"
        [2] OCTETSTRING = "STR Manor Park's registration"
[7] SET
 (P,C,43) LEGACY_SOURCE
   SET
    SET
     (P,C,40) EHCR-SOURCE
       SET
         [1] OCTETSTRING = "MiniClinic"
[8] SET
 (P,C,41) EHCR
   SET
    [5] SET
     (P,C,45) PATIENT-VERSION
       SET
        [0] (P,C,72) PATIENT
          SET
           (P,C,66) PERSON
             SET
              (P,C,65) PROVIDER
                INTEGER = "6"
[0] (P,C,73) PERSON-NAME
 SET
  [2] (P,C,23) MULTI-TEXT
   SEQUENCE
    (P,C,21) EHCR-INFO
      SET
        OCTETSTRING = "test"
     [0] SEQUENCE
      (P,C,22) PLAIN-TEXT
       SET
        (P,C,21) EHCR-INFO
          SET
            OCTETSTRING = "test"
         [0] OCTETSTRING = "Mr"
  [0] OCTETSTRING = "Kevin"
        [1] OCTETSTRING = "Gordon"
        [1] OCTETSTRING = "Neville"
      [0] (P,C,56) DATE-TIME
        SET
          [0] OCTETSTRING = "28-12-1956"
      [1] ENUMERATED = "0"
[3] SET
 (P,C,10) ADMIN
   SET
    (P,C,2) STANDARD-TRANS
      SET
```

```
(P,C,0) VERSIONED-TRANS
 SET
  [0] OCTETSTRING = "T 2"
  [1] (P,C,56) DATE-TIME
   SET
    [0] OCTETSTRING = "08101997"
    [1] OCTETSTRING = "155805"
  [4] OCTETSTRING = "version1"
  [6] SET
(P,C,1) TRANS-VERSION
 SET
  [0] (P,C,81) VERSION
   SET
  [1] SET
   (P,C,18) HRI
    SET
     (P,C,16) OBSERVATION
      SET
       (P,C,14) EHCR-ENTRY
        SET
         [2] (P,C,24) TERM-REF
          SET
           (P,C,22) PLAIN-TEXT
            SET
             (P,C,21) EHCR-INFO
              SET
               OCTETSTRING = "test"
             [0] OCTETSTRING = "MicroDoc Registration
Number"
    [0] OCTETSTRING = "O 3"
  [0] INTEGER = "3"
  [6] SET
   (P,C,33) QTY
    SET
     (P,C,30) QUANTITY
      SET
       [0] INTEGER = "1939"
(P,C,18) HRI
SET
 (P,C,16) OBSERVATION
  SET
   (P,C,14) EHCR-ENTRY
    SET
     [2] (P,C,24) TERM-REF
      SET
       (P,C,22) PLAIN-TEXT
        SET
         (P,C,21) EHCR-INFO
          SET
           OCTETSTRING = "test"
```

```
                    [0] OCTETSTRING = "Hospital Number"
                  [3] INTEGER = "1"
              [0] OCTETSTRING = "O 4"
          [0] INTEGER = "4"
          [6] SET
           (P,C,24) TERM-REF
            SET
             (P,C,22) PLAIN-TEXT
              SET
               (P,C,21) EHCR-INFO
                SET
                 OCTETSTRING = "test"
                [0] OCTETSTRING = "433488"
      (P,C,18) HRI
       SET
        (P,C,16) OBSERVATION
         SET
          (P,C,14) EHCR-ENTRY
           SET
            [2] (P,C,24) TERM-REF
             SET
              (P,C,22) PLAIN-TEXT
               SET
                (P,C,21) EHCR-INFO
                 SET
                  OCTETSTRING = "test"
                 [0] OCTETSTRING = "Hospital Name"
               [3] INTEGER = "1"
            [0] OCTETSTRING = "O 5"
        [0] INTEGER = "5"
        [6] SET
         (P,C,24) TERM-REF
          SET
           (P,C,22) PLAIN-TEXT
            SET
             (P,C,21) EHCR-INFO
              SET
               OCTETSTRING = "test"
              [0] OCTETSTRING = "Northern General Hospital"
      (P,C,18) HRI
       SET
        (P,C,16) OBSERVATION
         SET
          (P,C,14) EHCR-ENTRY
           SET
            [2] (P,C,24) TERM-REF
             SET
              (P,C,22) PLAIN-TEXT
               SET
                (P,C,21) EHCR-INFO
```

```
                    SET
                      OCTETSTRING = "test"
                    [0] OCTETSTRING = "Occupation"
                  [3] INTEGER = "1"
              [0] OCTETSTRING = "O 6"
          [0] INTEGER = "6"
          [6] SET
           (P,C,24) TERM-REF
            SET
             (P,C,22) PLAIN-TEXT
                SET
                 (P,C,21) EHCR-INFO
                  SET
                    OCTETSTRING = "test"
                  [0] OCTETSTRING = "Unemployed"
(P,C,13) CONTACT
 SET
  [0] (P,C,56) DATE-TIME
   SET
    [0] OCTETSTRING = "18-01-1994"
  (P,C,2) STANDARD-TRANS
   SET
    (P,C,0) VERSIONED-TRANS
     SET
     [0] OCTETSTRING = "T 12"
     [1] (P,C,56) DATE-TIME
      SET
       [0] OCTETSTRING = "08101997"
       [1] OCTETSTRING = "155809"
     [4] OCTETSTRING = "version1"
     [6] SET
      (P,C,1) TRANS-VERSION
       SET
        [0] (P,C,81) VERSION
         SET
        [1] SET
         (P,C,18) HRI
          SET
           (P,C,16) OBSERVATION
            SET
             (P,C,14) EHCR-ENTRY
              SET
               [2] (P,C,24) TERM-REF
                SET
                 (P,C,22) PLAIN-TEXT
                  SET
                   (P,C,21) EHCR-INFO
                    SET
                     OCTETSTRING = "test"
                    [0] OCTETSTRING = "Hypoglycaemia"
```

```
                [3] INTEGER = "1"
            [0] OCTETSTRING = "O 13"
        [0] INTEGER = "13"
        [6] SET
         (P,C,24) TERM-REF
          SET
           (P,C,22) PLAIN-TEXT
            SET
             (P,C,21) EHCR-INFO
              SET
               OCTETSTRING = "test"
                [0] OCTETSTRING = "No"
    (P,C,18) HRI
     SET
      (P,C,16) OBSERVATION
       SET
        (P,C,14) EHCR-ENTRY
         SET
          [2] (P,C,24) TERM-REF
           SET
            (P,C,22) PLAIN-TEXT
             SET
              (P,C,21) EHCR-INFO
               SET
                OCTETSTRING = "test"
                 [0] OCTETSTRING = "Therapy Change"
                [3] INTEGER = "1"
            [0] OCTETSTRING = "O 14"
        [0] INTEGER = "14"
        [6] SET
         (P,C,24) TERM-REF
          SET
           (P,C,22) PLAIN-TEXT
            SET
             (P,C,21) EHCR-INFO
              SET
               OCTETSTRING = "test"
                [0] OCTETSTRING = "No"
    (P,C,18) HRI
     SET
      (P,C,16) OBSERVATION
       SET
        (P,C,14) EHCR-ENTRY
         SET
          [2] (P,C,24) TERM-REF
           SET
            (P,C,22) PLAIN-TEXT
             SET
              (P,C,21) EHCR-INFO
               SET
```

```
                        OCTETSTRING = "test"
                      [0] OCTETSTRING = "Weight"
                    [3] INTEGER = "1"
                  [0] OCTETSTRING = "O 15"
                [0] INTEGER = "15"
                [6] SET
                 (P,C,36) QTY-WITH-UNITS
                  SEQUENCE
                   (P,C,33) QTY
                    SET
                     (P,C,30) QUANTITY
                      SET
                       [0] REAL = "131.2"
              (P,C,35) Q-WITH-UNITS
               SET
                [0] SEQUENCE
                 SEQUENCE
                  (P,C,58) UNIT
                   SET
                    [0] (P,C,24) TERM-REF
                     SET
                      (P,C,22) PLAIN-TEXT
                       SET
                        (P,C,21) EHCR-INFO
                         SET
                          OCTETSTRING = "test"
                        [0] OCTETSTRING = "kg"
                      [1] BOOLEAN = "-1"
            (P,C,18) HRI
             SET
              (P,C,16) OBSERVATION
               SET
                (P,C,14) EHCR-ENTRY
                 SET
                  [2] (P,C,24) TERM-REF
                   SET
                    (P,C,22) PLAIN-TEXT
                     SET
                      (P,C,21) EHCR-INFO
                       SET
                        OCTETSTRING = "test"
                      [0] OCTETSTRING = "systolic BP"
                    [3] INTEGER = "1"
            [6] (P,C,15) HEADING
             SET
              (P,C,14) EHCR-ENTRY
               SET
                [0] INTEGER = "16"
                [1] OCTETSTRING = "E 16"
                [2] (P,C,24) TERM-REF
```

```
                    SET
                     (P,C,22) PLAIN-TEXT
                      SET
                       (P,C,21) EHCR-INFO
                        SET
                         OCTETSTRING = "test"
                        [0] OCTETSTRING = "Blood Pressure"
                      [3] INTEGER = "1"
    [0] OCTETSTRING = "O 17"
[0] INTEGER = "17"
[6] SET
 (P,C,36) QTY-WITH-UNITS
  SEQUENCE
   (P,C,33) QTY
    SET
     (P,C,30) QUANTITY
      SET
       [0] INTEGER = "154"
    (P,C,35) Q-WITH-UNITS
     SET
      [0] SEQUENCE
       SEQUENCE
        (P,C,58) UNIT
         SET
          [0] (P,C,24) TERM-REF
           SET
            (P,C,22) PLAIN-TEXT
             SET
              (P,C,21) EHCR-INFO
               SET
                OCTETSTRING = "test"
              [0] OCTETSTRING = "mmHg"
        [1] BOOLEAN = "-1"
(P,C,18) HRI
 SET
  (P,C,16) OBSERVATION
   SET
    (P,C,14) EHCR-ENTRY
     SET
      [2] (P,C,24) TERM-REF
       SET
        (P,C,22) PLAIN-TEXT
         SET
          (P,C,21) EHCR-INFO
           SET
            OCTETSTRING = "test"
          [0] OCTETSTRING = "diastolic BP"
      [3] INTEGER = "1"
[6] INTEGER = "16"
[0] OCTETSTRING = "O 18"
```

```
[0] INTEGER = "18"
[6] SET
 (P,C,36) QTY-WITH-UNITS
  SEQUENCE
   (P,C,33) QTY
    SET
     (P,C,30) QUANTITY
      SET
       [0] INTEGER = "96"
    (P,C,35) Q-WITH-UNITS
SET
 [0] SEQUENCE
  SEQUENCE
   (P,C,58) UNIT
    SET
     [0] (P,C,24) TERM-REF
      SET
       (P,C,22) PLAIN-TEXT
        SET
         (P,C,21) EHCR-INFO
          SET
           OCTETSTRING = "test"
         [0] OCTETSTRING = "mmHg"
 [1] BOOLEAN = "-1"
```

References

[ASN188] Abstract Syntax Notation one - the primary tool for the specification of international standards - see: ISO 8824 'Specification of ASN.1' and ISO 8825 'Specification of Basic Encoding Rules for ASN.1'.

[Brya98] Bryan, M. "Guidelines for using XML for Electronic Data Interchange" Version 0.05 http://www.xmledi.net

[Cair91] Cairns T., Timimi H., Thick M., Gold G., "A generic model of clinical practice - *the Cosmos Project.*" *In: Lecture Notes in Medical* Informatics 45 (eds. Adlassnig K-P, Grabner G, Bengtsson S, Hansen R.) Pubs. Springer-Verlag 1991

[Catt93] Cattermole D. "SHAPSM - Sheffield Health Authority Prototype Structured Message. Electronic Communication for the Exchange of Diabetes Information." Msc Thesis University of Hull 1993

[Cham94] The Chambers Dictionary, Chambers published 1994.

[Coch96] Cochrane P: *(Invited Keynote Paper)* "Don't *Silicomorphise Me!"* Human Computer Interaction Conference (HCI 96), Imperial College, London, 20 - 23, August 96, pp1-6. Also given at "The John Jones Memorial Lecture" University of Hull, October 1996.

[Comm94] The "Snn proposal" - a short term solution to the segment group trigger problem. The EC UN/EDIFACT co-ordinator, Commission of the European Communities.

[Comp96] *"Dictionary of Computing Forth Edition" Oxford University Press 1996*

[CR1300] CR1300 Report. Published by CEN TC251 WGI 1992.

[Dixo97a] Dixon R. "The Good European Health Record Object Model", Towards an Electronic Patient Record '97

[Dixo97b] Dixon R, Lloyd.D. "Proposal for GOM v1.5" – Internal Document for EHCR-SupA,1997

[Dixo98] Dixon R, Grubb P, Camplin D. "So Sue Me!" proceedings of Making Medical Informatics Work, February 1999. British Medical Informatics Society, Manchester Metropolitan University

[Dixo99] Dixon R, Heller S, Harris N, Camplin D, Grubb P, "The SHINDIG Project: Some lessons in implementing the GEHR architecture" Health Informatics Journal November 1999 in press. Published by Sheffield Academic Press.

[Domb96] de Dombal F.T. "Medical Informatics..... the essentials" Chapter 7. Butterworth Heinemann 1996, ISBN 0 7506 2162 1

[Dohl92] Department of Health London. Executive Letter EL (92) 34. Adoption of UN/EDIFACT as the NHS Standard for Electronic Data Interchange of structured messages 1992.

[Dude98] Markwell, D., Dudeck, J. "Report from Task Force XML" CEN/TC 251/N98-080. 4th September 1998.

[Eced91] EDIFACT Training Manual, Commission of the European Communities 1991

[Eimg94] Approved Working Documentation for the v1.0 trial NHS standard EDIFACT message for Pathology Requests and Reports, Volume 1 NHS Executive Information Management Group

[Elli96a] Ellis J.C., Dixon R.M., Grubb P.A., 'The EDIFACT standard NHS Pathology Message is not Standard EDIFACT' Medical Informatics Group Report Series - 1996 ISBN: 0 85958 911 0

[Elli96b] Ellis J.C., et al, "Health Care Data Exchange for the 21st Century", proceedings of Toward an Electronic Health Record Europe '96, Nov 1996, pp 251-255 ISBN 0 9640667 85

[Elli97] J.C.Ellis, R.M.Dixon, P.A.Grubb, D.A.Camplin, 'The Integration of data in existing systems with systems of the future', Proceedings from Toward An Electronic Patient Record 1997, Nashville, Tennessee, USA. ISBN 0-9640667-9-3

[Exch94] Draft British Standard Implementation of prEN 1613. Medical Informatics - Messages for exchange of Laboratory information. Document 94/647884, IST/35

[Feat98] Featherstone H., "The next generation of EDI standards and the future direction of Accredited Standards Committee (ASC) X12" Strategic Implementation Task Group September, 1998.

[Frat94] Traducteur EDIFACT Portable, France Telecom, L'Equerre 7-9, rue Hélène Boucher, Saint-Quentin-en-Yvelines 78285 Guyancourt Cedex, France .

[Gang92] Gangemi A., Galanti M., Galeazzi E., Ross Mori A. "Beyond UMLS: computational semantics for medical records", MEDINFO 92.

[GEHR92] "GEHR Requirements for Clinical Comprehensiveness" (GEHR Project Deliverable 4, 144 pp), St. Bartholomews Hospital Medical College, 1992.

[GEHR93a] "GEHR Requirements for Portability" (Project Deliverable 5, 141 pp), St. Bartholomews Hospital Medical College, 1993.

[GEHR93b] "GEHR Requirements for Communication" (Project Deliverable 6, 139 pp), St. Bartholomews Hospital Medical College, 1993.

[GEHR93c] "Clinical Functional Specifications" (Project Deliverable 7), St. Bartholomews Hospital Medical College, 1993

[GEHR94a] "Ethical and Legal Requirements of GEHR Architecture and Systems" (Project Deliverable 8, 69 pp), St. Bartholomews Hospital Medical College, 1994.

[GEHR94b] Educational Requirements of GEHR Architecture and Systems (Project Deliverable 9, 62 pp), St. Bartholomews Hospital Medical College, 1994.

[GEHR95] GEHR Deliverables 19,20,24 – "GEHR Architecture", Version 1.0, 30/6/95, D. Ingram, D. Lloyd, D. Kalra, T. Beale, S. Heard, P. A. Grubb, R. M. Dixon, D. A. Camplin, J. C. Ellis, A. M. Maskens.

[Grah94] Graham,I. "EDI Impact: Social & Economic Impact of Electronic Data Interchange (EDI)". Report prepared for the European Commission. University of Edinburgh. TEDIS Project report C9. http://www.ed.ac.uk/~ehja36/tedis_c9.html

[Grim87] Grimaldi R.P. "Discrete and combinatorial mathematics, An applied introduction" Addison-Wesley 1987

[Grim96] Grimson J. et al, "SYNAPSES Federated Health Care Record Server", Medical Informatics Europe '96, IOS Press, 1996 pp 695-699

[Grub91] Grubb, P.A. "The Impact of Information Technology upon Primary Health Care in Great Britain" The Brynmor Jones Library, The University of Hull. Q 1992 Ph. D. G8

[Grub93] Grubb P.A. and Dixon R.M. "The communication of diabetic patient information through Information Technology" The Forum on Computers in Medicine. Chester April 1993.

[Grub97] Grubb P. et al, "Specification Document for Data Transfer from/to non-GEHR-based Sytems using GEF" - in press, 1997

[Hawk95] Hawking M. "Code Conversions, Data Stability and the Future - an Agenda for Discussion", Journal of Informatics in Primary Care, BCS, Sept. 95, pp. 3-5.

[Hinc98] Hinchley, A. "Short strategic study: Enabling Technologies – SGML/XML (final report) CEN/TC 251/N98-061, 16th July 1998.

[HL7-96] "Health Level Seven Specification, Section 1.1" Version 2.3 ©1996, Membership ballot #2

[Holl94] Holland W. et al, "Back to the future Public health research into the next century", Journal of Public Health Medicine March 1994

[Hosp95] "State of the art in European Healthcare EDI and recommendations for further development" March 1995. EDIFORHEALTH Project. Consorsi Hospitalari Catalunya.

[Im&T94] "A strategy for NHS-wide networking" NHS executive. Published by the Information Management Group of the NHS Executive. August 1994, IMGE E5155.

[Inve93] "Investigation of syntaxes for existing interchange formats to be used in healthcare" Published by CEN TC251 WGI. CR1350

[Kalr96] Kalra D., Lloyd D. "The Synapses Object Model and Object dictionary"

in the proceedings of 'Toward an Electronic Health Record Europe 96' ISBN 0 9640667 85

[Kay93] Kay J., McVittie J. "Communication of Structured Medical Information by Computers: The Oxford Experience" Progress in Standardisation in Health Care Informatics G.J.E De Moor et al. (Eds.) IOS Press,1993

[Lang98] Langlands A., "Information for Health: An Information Strategy for the Modern NHS (HSC 1998/168)". Published by The Department of Health 1998

[Mars96] Marsden, B.W. "Communication Network Protocols" 2nd Edition. Pp.188-189 1996 ISBN 0-86238-106-1

[Mila96] Milan J. et al, "Electronic Patient Records: The Need for an Underlying Structure", proceedings of Toward an Electronic Health Record Europe '96, Nov 1996, pp141-148 ISBN 0 9640667 85

[Neuf92] Neufeld.G, Vuong,S. "An overview of ASN.1" Computer Networks and ISDN Systems,1992 Vol23,No 5 pp393-415

[Priv94] Private communication

[PT26] "Health Informatics – Electronic Healthcare Record Communication – Part 1: Extended Architecture and Domain Model" CEN TC251 PT26

[PT27] "Health Informatics – Electronic Healthcare Record Communication – Part 2: Domain Term Lists" CEN TC251 PT27

[PT28] "Health Informatics – Electronic Healthcare Record Communication – Part 3: Distribution Rules" CEN TC251 PT28

[PT29] "Health Informatics – Electronic Healthcare Record Communication – Part 4: Messages for the Exchange of Record Information" CEN TC251 PT29

[Rect91] Rector AL, Nowlan WA and Kay S. "Foundations for an Electronic Medical Record". Methods of Information in Medicine, 1991; 30: pp 179-186, FK Schattaeur Verlagsgesellschaft mbH publications. Also reprinted in JH van Bemmel and AT McCray (eds), IMIA Yearbook of Medical Informatics 92, Advances in an Interdisciplinary Science,

Schatteur Publications, 1992, pp 59-66.

[Rect95] Rector AL, Zanstra P, Solomon WD., "GALEN: Terminology Services for Clinical Information Systems" Published in MF Laires MJ Ladeira and JP Christensen (eds) Health in the New Communications Age,: Health care telematics for the 21st century. IOS Press, Amsterdam. pp90-100.

[RFA95] NHS Executive General Medical Practice Computer Systems – "Requirements for Accreditation Version 3",1995/96

[Rodr97] Rodrigues, J-M. et al. "Galen-in-use: An E.U. project applied to the development of a new national coding scheme for surgical procedures: NCAM" Medical Informatics Europe '97 pp897-901 (Thessaloniki, Greece) IOS Press Vol 43 ISSN:0926-9630.

[Ross95] Anderson R., "NHS Wide Networking and Patient Confidentiality" Senior Research Associate, Computer Security Group, University of Cambridge.

[Rouv94] TS_TREDI manual de Référence (1) TRA/DEV TS 001 Version 1.5 J.Rouvière. 4th January 1994

[Rule93] UN/EDIFACT Message Guidelines and Rules. The EC UN/EDIFACT co-ordinator, Commission of the European Communities.

[Samp93] Snacc 1.1: A high performance ASN.1 to C/C++ compiler by Micheal Sample. Dept. Computer Science, University of British Colombia. Feb 1993, Updated July 1993

[Sitp92] Message Design Guidelines, Reference Documentation, SITPRO EDIFACT service, SITPRO 1992.

[Tded93] Trade Facilitation, Trade Data Elements Directory, UNTDED 1993. ISO 7372:1993 ISBN 92-1-116585-7.

[Tele97] NHS Telecommunications Branc "NHSnet: A Briefing for NHS Chief Executives Contributor: NHS Telecommunications Branch April 30, 1997. Published by the NHS Information Authority © Crown Copyright 1999 NHS IM&T Electronic Library: Catalogue Card 326

[TMWG] "Techniques and Methodologies Working Group Frequently Asked Questions", Techniques and Methodologies Working Group Information. http://www.harbinger.com/resource/klaus/tmwg/faq.html

[Tous96] Toussaint P. J. et al, "Supporting Shared Care for Diabetic Patients", proceedings of Toward an Electronic Health Record Europe '96, Nov 1996, pp 117-121 ISBN 0 9640667 85

[Wats89] Watson D. "High-Level Languages and Their Compilers" Addison Wesley ISBN 0-201-18489-3