

**Good European Health Record
(EC AIM GEHR A2014)**

Document identifier: Deliverables 19,20,24

Version number: 1.0

Document title: GEHR Architecture

Workpackage activity ref: All

Approval status: Final

Date of issue: 30.6.95

Contact name: David Lloyd

Address: Centre for Health Informatics and
Multiprofessional Education
University College London
Whittington Hospital Campus
London N19 5NF

Telephone: +44 171 288 3364

Fax: +44 171 288 3322

Email: d.lloyd@chime.ucl.ac.uk

URL: <http://www.chime.ucl.ac.uk/HealthI/GEHR/deliverables.htm>

Number of pages = 250

Version 1.0

Contributors to the writing of this Document

Coordinating Partner

Professor David Ingram
Mr David Lloyd
Dr Dipak Kalra
Mr Thomas Beale
Dr Sam Heard

University of Hull

Dr Penny Grubb
Dr Richard Dixon
Dr David Camplin
Mr Jeremy Ellis

Health Data Management Partners

Dr Alain Maskens

TABLE of CONTENTS

1	Introduction	7
1.1	The Overall Approach of the GEHR Project	7
1.2	The Approach to Developing the Architecture	8
1.3	Overview of this Deliverable	9
2	Background	11
2.1	Concepts and Terminology	11
2.2	Project Methodology	12
2.3	Boundaries of the Electronic Healthcare Record Domain	14
3	Summary of User Requirements	15
3.1	Introduction	15
3.2	Summary of the Requirements for Clinical Comprehensiveness ...	16
3.2.1	The Context of Clinical Records	16
3.2.2	The Historical Background of Clinical Records	18
3.2.3	Clinical Competence and the Clinical Record	19
3.2.4	The Role of the Clinical Record	20
3.2.5	The Nature of Clinical Data	21
3.2.6	The Structure of Medical Language	22
3.2.7	Evolving From Paper Records	23
3.2.8	The Importance of Data Security	25
3.2.9	Drug Prescription and Its Incorporation Within the Clinical Record ..	25
3.2.10	The Role of the Clinical Record in Education	25
3.2.11	Summary of Research Investigations Undertaken	26
3.2.12	Overall Conclusions	29
3.3	Summary of the Ethical and Legal Requirements	30
3.4	Summary of the Educational Requirements	31
3.4.1	Introduction	31
3.4.2	Preparing Healthcare Professionals to Manage Patient Information ...	32
3.4.3	How Electronic Records could Benefit Education	32
3.4.4	Requirements for the GEHR Architecture to Support the Use of Electronic Health Records in an Educational Environment	32
3.5	Summary of the Clinical Functional Specification	34
3.5.1	Introduction	34
3.5.2	The Transaction	34
3.5.3	Transaction Types	35
3.5.4	The Transfer of Healthcare Records	37
3.5.5	The Confidentiality of Healthcare Data	37
3.5.6	Making Use of a Shared Clinical Record	37
3.5.7	Viewing a Clinical Record	38
3.5.8	Confirming which language has been used	38
3.5.9	Confirming which classification system has been used	38
3.5.10	Recording a Transaction	39
3.5.11	Recording clinical data	39
3.5.12	Protocols, templates and decision support	40
3.5.13	Local term sets and units	40

3.5.14	Amending a Transaction	41
3.5.15	Death of a Patient	41
4	Overview of the GEHR Architecture.....	43
4.1	Background	43
4.2	Summary of Principal Architectural Components	44
4.3	Description of the Principal Architectural Components	44
4.3.1	EHCR.....	44
4.3.2	Transaction	44
4.3.3	The Organisation of Healthcare Data within Transactions.....	47
4.3.4	Health Record Item	49
4.3.5	Health Record Item Collection	51
4.3.6	Heading	52
4.3.7	Attributes	53
4.3.8	The Choice of Constructs to represent Clinical data.....	53
4.3.9	Effects of Available Termsets on the Choice of Constructs	55
5	GEHR Object Model Classes	59
5.1	EHCR Cluster.....	59
5.1.1	Acquisition Requests	60
5.1.2	Access and Security	60
5.2	Identification of Persons in the EHCR.....	61
5.3	Transaction Cluster	62
5.3.1	Version Control.....	62
5.3.2	VERSIONED_TRANS	64
5.3.3	TRANS_VERSION	65
5.3.4	ADMIN	65
5.3.5	CONT_CARE	66
5.3.6	CONTACT	66
5.3.7	NOTA_BENE	66
5.3.8	REPORT.....	66
5.3.9	SUMMARY.....	66
5.3.10	TRIGGER	66
5.4	Item Cluster	67
5.4.1	HEADING.....	71
5.4.2	HRI	71
5.4.3	OBSERVATION.....	72
5.4.4	HRI_COLLECTION	72
5.5	EHCR_info Cluster	73
5.5.1	EHCR_INFO	73
5.5.2	Generic Data Types	73
5.6	Text Cluster	73
5.6.1	TERMSET_DESC	75
5.6.2	TERM_SET.....	75
5.7	Quantity Cluster	75
5.7.1	QUANTITY.....	75
5.7.2	QTY_RATIO	76
5.7.3	Q_RANGE	76
5.8	Units Cluster.....	76
5.9	Bulky Data Cluster.....	77
5.10	People and Places Cluster	78
5.11	Moment Cluster.....	79
5.12	GEHR Basic Cluster.....	79

5.13	Exchange Cluster	79
5.13.1	Universal Resource Identifier (URI)	79
5.13.2	GEHR_UID	80
5.13.3	REVISION_ID	81
6	Formal Expression of the GEHR Architecture.	83
6.1	Introduction	83
6.2	Object Modelling Formalism Used	84
6.2.1	Diagram Conventions	86
6.3	GEHR Object Model Diagrams and Class Texts	90
7	Communicating Data.	155
7.1	Minimum requirements.	155
7.2	Extracting data for transmission	155
7.2.1	GEHR-Compliant systems	155
7.2.2	Legacy Systems.	155
7.3	Receiving data.	156
7.4	The GEHR Exchange Format.	156
8	The GEHR Term Set.	167
8.1	Introduction	167
8.1.1	Initial Clinical Evaluations	167
8.1.2	The Final GEHR Term Set	168
8.2	Semantic Issues	169
8.2.1	The Decomposition of Terms	169
8.2.2	Synonyms	171
8.2.3	Plurality	171
8.2.4	Gender and Plurals.	172
8.2.5	Laterality.	172
8.2.6	Other Anatomical Sites	173
8.2.7	Other Qualifying Terms.	174
8.2.8	Conclusions	174
8.3	Evolution of Term Sets	174
9	Implementation and Tools	177
9.1	Implementation Aspects	177
9.1.1	Libraries and Applications	177
9.1.2	Data Storage Implementations.	178
9.1.3	Disclaimer	181
9.2	Tools	181
9.2.1	The Clinical Drawing Tool.	183
9.2.2	Image Tool.	184
9.2.3	Mailbox Server Mechanism	185
9.2.4	GEHR Upgrade Tool	185
9.2.5	Communication with Other Systems and Standards.	186
9.2.6	Conversion Tools.	187
9.2.7	Summary.	189
10	The Way Forward	191
10.1	The Future	191
10.1.1	Anticipated Areas of Future Development	193
10.1.2	Future Standardisation: Relationship between GEHR and CEN	194
10.1.3	GEHR Authority	195
11	References.	199

Appendices

A	Information and Object Modelling	201
A.1	Requirements of a Formal Model.	203
A.1.1	Validateable	203
A.1.2	Definition of Ideas	203
A.1.3	Communication of Ideas	203
A.1.4	Implementable.	204
A.1.5	Conformance Testing	204
A.1.6	Completeness	204
A.2	Object Models	205
A.2.1	Semantics Available in an Object Model (OM)	205
A.2.2	Seamlessness.	206
A.2.3	Extendible at any Level	207
A.2.4	Naming	207
A.2.5	Modelling Incomplete Concepts.	208
A.3	General Approach to Modelling.	209
A.3.1	Comments on Modelling Formalism Chosen	209
A.3.2	Minimum Features	210
A.3.3	Function Features	210
A.3.4	Correctness	211
A.3.5	Classification of Features	211
A.3.6	Where does the Model Stop?	211
A.4	Object References	213
A.4.1	Meaning	213
A.4.2	Implementation	213
A.4.3	Persistent Objects	214
A.4.4	Exchange of Objects	214
A.4.5	Persistent and Exchangeable Objects	215
A.5	Special Actions and Data.	217
A.5.1	Term Sets	217
A.5.2	Querying	217
A.5.3	External Information.	217
A.5.4	Reference Data	218
A.5.5	Bulky Data	219
A.5.6	Multi Media	219
A.5.7	Clinical Test Results	219
A.5.8	Comment.	219
B	A Worked Example of the Application of the GEHR	
	Architecture to a Clinical Situation.	221
B.1	Scenario	221
B.2	Outline of the Structures Used	221
B.3	Detailed description of Classes and Attributes	225
C	Other Considerations	245
C.1	Stages in the Evolution of the GEHR Architecture	245
C.2	Future Work.	247
C.3	Minimum and Compromise Solutions.	247
C.4	Results of Others' Work	248

C.5	Particular Solutions Chosen.....	248
C.6	Ideas Rejected as Inadequate or Wrong.....	248
C.7	Outside the record.....	249
D	Glossary of Acronyms.....	250

List of Figures

FIGURE 1	Methodology	13
FIGURE 2	The Context of the Clinical Record	17
FIGURE 3	The Transaction	46
FIGURE 4	The Health Record Item (HRI)	50
FIGURE 5	The HRI Collection	52
FIGURE 6	EHCR with Transactions and Version Control	63
FIGURE 7	EHCR_ENTRY General Structure	68
FIGURE 8	HRI and Collection Example with Headings	70
FIGURE 9	Abridged GEHR Object Model	85
FIGURE 10	EHCR Cluster	90
FIGURE 11	Transaction Cluster	94
FIGURE 12	EHCR Item Cluster	102
FIGURE 13	EHCR Info Cluster	108
FIGURE 14	Text Cluster	110
FIGURE 15	Quantity Cluster	117
FIGURE 16	Units Cluster	124
FIGURE 17	Bulky Data Cluster	129
FIGURE 18	Moment Cluster	134
FIGURE 19	Persons and Places Cluster	137
FIGURE 20	GEHR Basic Cluster	143
FIGURE 21	GEHR Exchange Cluster	147
FIGURE 22	Underlying Information vs Views	202
FIGURE 23	Inheritance, Association and Aggregation Example ..	205
FIGURE 24	Object Model Seamlessness	206

1 Introduction

1.1 The Overall Approach of the GEHR Project

Clinicians are becoming increasingly aware of the opportunities the computer offers in support of clinical practice. The management of complex diseases, clinical audit, and the automatic generation of reports are examples. The growing complexity of healthcare services means that managers need greater access to aggregate information about the processes of clinical care. Unfortunately, the computer systems currently used in most hospitals and general practice surgeries, and more importantly the data modelling concepts which underlie them, are ill-equipped to cope with these new challenges.

The Good European Healthcare Record (GEHR) project was established within the EU Telematics Research and Technology Development Programme to develop a comprehensive and widely applicable common data structure for using and sharing electronic healthcare records within Europe. The information environment in which such an architecture might be used includes all sites capable of creating and maintaining clinical and clinically-related data. The user organisations which might become involved range in size from large health regions and hospital groups with dedicated computing departments to single handed practices, each with a PC and a modem. The technical infrastructure of such organisations encompasses many different technologies of database, computer system and network, as well as many levels of software engineering capability. The commercial organisations developing and supporting products within this infrastructure are similarly diverse; computer and systems vendors, large and small software companies; health and management consultancies. [1]

The GEHR Architecture provides a framework which can support the full diversity of clinical data storage, retrieval and communication required by clinicians. It has been formulated to encompass the requirements of the different disciplines of primary and secondary health care, for use by doctors, nurses and other professionals in all European countries. The ready access to a wide range of data and data-types is of increasing clinical importance and the work of the project has included these new multimedia aspects of the record. Examples specifically addressed include x-ray and photographic images, bio-signals, clinical drawings as well as, most importantly, textual information. The latter includes, for example, clinical observations and laboratory data in the form of coded terms and free text. The architecture provides a framework upon which systems may be built to support the core ethical and legal requirements of good clinical practice.

The priorities of the GEHR project have reflected the belief that the clinical record is most necessary, and should be most available, when a clinician is offering care in a consultation or at the bedside. Thus efforts should always be directed towards offering quickly accessible, accurate and complete information to an authorised carer when attending a patient.

The clinical record must be resilient to the challenges arising in three areas of change: in time, place and clinical perspective. A healthcare record evolves gradually over a person's lifetime and family records over generations. A person's healthcare needs will change and evolve in time, as does the practice of medicine, and the economic and social framework within which medicine is practised and managed. The clinical record will be used by staff trained in different disciplines, working in different settings, on different sites and using different languages. It must

accommodate evolving needs for coding and classification standards and for the use of clinical guidelines in the management of care.

These diverse contexts of care are reflected in important differences in personal and professional practice. The early stages of the project were devoted to exploration and identification of these user requirements, classified into areas of clinical comprehensiveness, portability, communication, ethical and legal acceptability and education. A key issue confronted within the GEHR project was to determine at what level it was feasible and appropriate to standardise the structure of the information contained within the record and the exchange of that information, while satisfying clinical needs. [2]

1.2 The Approach to Developing the Architecture

GEHR has brought together a large, clinically-centred but multi-professional project team. Its members have come from industrial, healthcare and academic backgrounds and from diverse countries of Europe. Exploring and reaching conclusions about the diversity and the commonality of healthcare needs and practices in this uniquely committed and inspiring, but sometimes difficult grouping has been a challenge for all. By seeking, as the highest principle uniting the group, to keep issues alive and under debate; by seeking never to allow *force majeure* from one side or another to prevail in deciding issues of principle; and with the benefit of broad consultation and research in and beyond AIM, much progress has emerged.

The Project team concluded, and GEHR then worked on the premise, that what needs to be standardised is not clinical practice *per se*, but rather the semantics of the information documented in the process of providing clinical care. Within the context of healthcare information there is a boundary between what needs to be standardised and what does not. The GEHR project has sought to define that boundary in order to identify the features which the architecture must provide as the basis of a structured electronic healthcare record.

To be useful, an architecture must facilitate the evolution of existing systems as well as the construction of new systems. It has to go beyond a lowest common denominator which attempts to satisfy everyone (thereby incorporating little real semantic content) but to stop short of what might become a semantically powerful standard, encapsulating the requirements of the healthcare community, which would only in practice be achievable on a green field site or after wholesale and uneconomic re-engineering of current systems.

The fundamental constructs of the GEHR architecture have been evolved by clinicians and computer-scientists to meet the general user requirements which the project has identified. They arise from and are centred in the team's understanding of those requirements and their implications for a formal architecture of the electronic healthcare record.

The GEHR architecture has been progressively refined using these evolving constructs, from early versions expressed largely in clinical language, through successive iterations and testing of prototypes, towards a formal and rigorous version. Early versions proved unacceptable in the face of progressively more detailed analytical scrutiny. The final version, based on the object-oriented approach and published here, has proven the best capable, thus far, of capturing the clinical requirements within a consistent formalism which permits greater clarity of definition, technical

and clinical validation and conformance testing. The GEHR architecture is presented here in terms of:

- **An Information Model, which provides a minimum specification for the structure and content of information held at a site**
- **An Exchange Format Specification, which specifies how information can thereby be exchanged between heterogeneous sites**

1.3 Overview of this Deliverable

This Deliverable documents the information model in detail (Chapters 4,5,6) and references the Exchange Format Specification (Chapter 7).

For the general reader, a summary is provided (Chapters 2,3) of the background to this work which has previously been extensively documented in the referenced Deliverables on user requirements, functional specifications, interim prototypes, demonstrations and evaluations. These Deliverables are available from the GEHR Project Coordinating Partner and should be consulted by those who require more detail of the work.

A key aspect of the dissemination of the work to a broad clinical and professional audience is the need for a persuasive set of illustrative examples. Such examples have been used throughout the GEHR Deliverables. A comprehensive clinical example is included in Appendix B to demonstrate application of the final version in a real situation. However, whereas the consistency and comprehensiveness of the information model may be tested, it is not possible to prove the applicability of the architecture in the clinical domain except through clinical validation where systems based on the architecture are tested against the functional specification for electronic records. There will be a large amount of clinical research involved in configuring these systems to capture and communicate records for specific clinical domains. Associated with this are all the evolving issues of the constraining of clinical vocabulary and the relationships within vocabularies which enable recording and analysis to be conducted accurately and efficiently.

The final results of the project will be published in the literature in due course. The GEHR partners have placed all of the work of the project in the public domain, reflecting their wish that it should be available openly and without constraint to support development in this complex but extremely important field. In this spirit it has been made fully available to the work of CEN TC/251 and to industry in Europe.

The architecture has been evolved to a considerable level of detail and some of this, along with the motivation and choice of formalism applied in the project, is dealt with in Appendices. The project has not found it easy to find and agree on formal methods to express its work but much progress has been made. Reflecting the status of this kind of work generally, no claim is made that the methods are yet either comprehensive or ideal, as discussed in Appendix C.

The rationale for the development of a GEHR term set is discussed in Chapter 8, together with the semantics issues which arose during its construction. The term set itself, translated into nine European languages, is included in this Deliverable as a text file on an accompanying floppy disk.

The GEHR architecture is not a healthcare record software system. Adoption of the GEHR architecture implies no assumptions about how data structures should be physically implemented within the databases of clinical records systems. The optimization of data storage and retrieval is a well developed technology. To assist implementors, some outline common guidelines and tools for implementation of systems conformant with the architecture have been developed and are covered in Chapter 9 and in Appendices. The intention is that the tools should be fully integrated with emerging GEHR compliant systems. Feedback from the assessment of both tools and systems at test sites will help to refine and extend the architecture.

One specific tool, for the creation of clinical drawings, is included as a demonstration software on the accompanying floppy disk. It is intended primarily to enable systems developers to browse a library of 50 anatomical drawings, designed by GEHR and which have been placed in the public domain. The full version of the clinical drawings software is copyright, and further enquiries about its use should be made to the Coordinating Partner.

Chapter 10 gives an indication of future developments envisaged if a common architecture is to be validated in clinical contexts to a point where it may achieve the status of an acceptable and accepted standard. In this context, a brief discussion is included on the relationship of the GEHR project with standards-making bodies such as CEN and the contribution of the project to such work on standardisation. Chapter 10 also briefly discusses the nature and evolution of term sets and their relevance to the architecture.

The Appendices provide further details of areas of future work. In addition, brief explanations are provided as to why certain decisions were made in developing the architecture and why particular possibilities were not pursued. Compromise solutions have been adopted in certain areas and these compromises are explained and justified in Appendix C.

References are in Chapter 11.

A glossary of acronyms is provided in Appendix D.

2 Background

This Chapter revisits material from earlier Deliverables of the project. First it summarises some general concepts and terminology fundamental to the discussion, to set the description of the Architecture in context [3]. A brief outline is given of the GEHR project methodology and the view adopted of the boundaries and scope of the electronic healthcare record.

2.1 Concepts and Terminology

Architecture

“Data are words and or numbers which have significant meaning. Data derive their meaning from a precise name and agreed upon definitions” [4].

“Information is data that have been organised and arranged to convey knowledge” [4]

It is the organisation of data which conveys knowledge. This organisation is called a structure, or an **architecture**.

Healthcare Record

“A record is an account of information or facts, set down in writing or pictorially as a means of preserving knowledge” [4]

Since much of modern health care takes place within teams of cooperating clinicians, this basic definition should be supplemented with: “a means of preserving *and communicating* knowledge”,

This, now extended, definition summarises the essence and purpose of the Healthcare Record (HCR) (which is expressed more fully in various requirements and functional specification documents documented in the first phase of the GEHR project - referenced below).

The Healthcare record is a documented account of information, with the purpose of preserving and communicating knowledge.

an account: what is kept in the record is not usually the facts but more modestly a description of the facts; the role of the person (system, process) recording the information is recognised here as an essential attribute of the record;

of information: the data and the way they are structured;

documented: collected, kept, stored in some way;

with the purpose of preserving and communicating knowledge: this clearly implies the specific purpose of the record.

Healthcare records are recorded accounts of information about a person's health related history. Such accounts are made and stored for each patient in each situation where health care is provided. On this basis, the healthcare record may be defined very simply in terms of its purposes and operations.

The main purpose of the healthcare record is, as indicated earlier, to preserve the information it contains. This information can be used for the management of patient care, for legal accountability, for research and statistics, for teaching, etc. [see the various requirements summarised below, Chapter 3].

The main operations made on or with the HCR are for data entry and storage, access and retrieval, editing, and transfer.

An *electronic* HCR (EHCR) needs to extend the basic definition with: “set down in writing, or pictorially, **or in *electronic format*.....”**

Healthcare Record Architecture

A healthcare record architecture is concerned not only with the data themselves but with the way they are organised so as to convey their full meaning. It is this complete structure which has been the specific domain of the GEHR project.

This information might be handled as free text, or within a progressively finer granularity of concepts and constructs, provided to capture the meaning and context of the data as recorded at that time and place and by a particular clinician.

Electronic Healthcare Record Architecture

The architecture for the *electronic* form of the Healthcare Record.

2.2 Project Methodology

A very specific methodology was adopted for the GEHR project, based upon the evolution and testing of prototypes within a comprehensive framework of requirements established across all sectors of health care.

The starting point for the series of prototypes was an existing health record architecture, the CHRA (developed by Health Data Management Partners, Brussels). The CHRA and a commercial product (Health.One) based upon it were extensively tested in English, Belgian, and French test sites and analysed by the partners in Luxembourg and Portugal.

The results of these tests were incorporated into the GEHR Deliverables 4-6 [5, 8, 9], concerned respectively with the requirements for clinical comprehensiveness, for portability, and for communication capacity. These Deliverables were also based on extensive reviews of literature

and of other medical management software, and on intensive discussion and research, within the GEHR working groups and with relevant professional bodies and organisations. The development of the GEHR architecture is underpinned by these requirements.

The clinical focus of the work was provided through the Requirements for Clinical Comprehensiveness [5], and later through the Requirements for Ethical and Legal Acceptability [10] and the Educational Requirements [11]. This essential clinical influence was further developed in the formulation of the Clinical Functional Specification, Deliverable 7 [7]. The latter specification, together with Deliverables 5 (Portability Requirements) and 6 (Communication Requirements) provided the foundation for a series of Interim GEHR Architectures (IGAs) which were used to explore and refine concepts and ideas.

The detailed development of prototypes served two main objectives:

- to allow for the design and implementation of software tools and testing in real life situations;
- to identify possible areas of uncertainty or ambiguity in the architecture.

The series of evolving prototypes was a key feature of the methodology, allowing all issues confronted within the project to be the subject of empirical testing at a set of implementation test sites. This proved a difficult but important discipline to chart the way towards a fully functional EHCR. It enabled realistic choices to be made as to what was currently achievable and implementable, what could be planned for in the medium term and what had to be recognised and accommodated as longer term objectives.

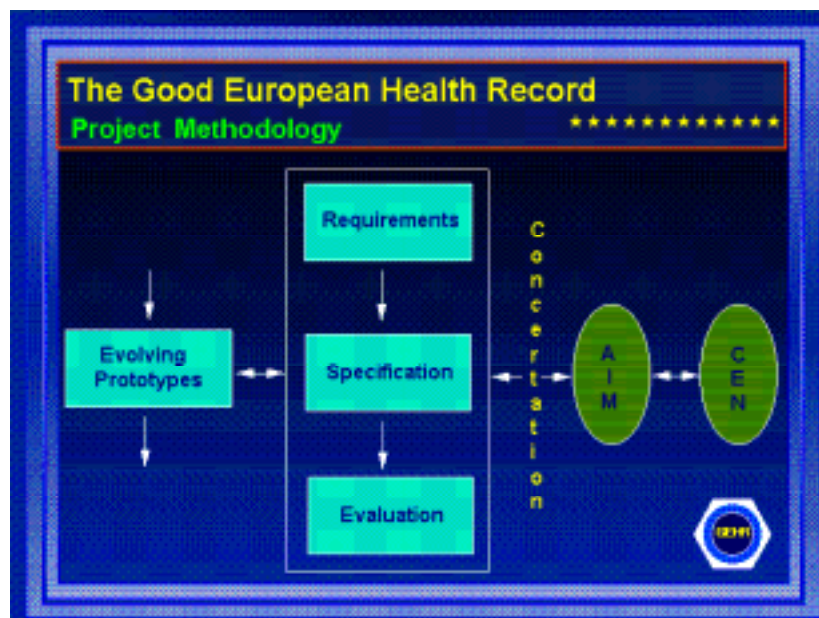


FIGURE 1 Methodology

2.3 Boundaries of the Electronic Healthcare Record Domain

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 latter approach arises from progress in the fields of database management systems and networking. With these 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 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 GEHR requirements [5]. For example:

- The rationale for clinical decisions must be apparent from the record (what was done and why).
- The results of investigations must be checked by the clinician in charge 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.

3 Summary of User Requirements

3.1 Introduction

The GEHR Architecture formalism was developed following an extensive investigation of clinical user and technical requirements and their subsequent incorporation into a functional specification. Doctors, nurses and other allied professions from across Europe were involved in deriving a set of requirements in several key areas:

- a the requirements for the comprehensive recording of consultations with patients for a wide range of disciplines in primary and secondary care, including the specific needs for a variety of multi-media data types, for the access to coded term-sets, and the requirement to accommodate both structured entries and free text;
- b the requirements for the portability of healthcare records between different institutional systems independent of the hardware configurations, of the operating systems or of the applications at those sites, and independent of the original language and of the term-sets used;
- c the requirements for the communication and amalgamation of healthcare records between clinicians involved in sharing the care of patients, whether via telecommunications networks or intermittently connected devices such as smart cards;
- d the ethical, medico-legal and security issues which arise when using EHCRs as the sole medium for recording and storing patient-related information, including both the features necessary within the Architecture and the regulatory framework which must support their legitimate use;
- e the educational needs at a pre- and post-registration level in order to enable the clinical work force to utilise these new technologies and to accommodate the necessary changes in the way healthcare information is managed.

These Deliverables, derived following extensive consultation and reviews of previous work in the field, provided the essential basis from which the architecture was developed. The progressive refinements to the architecture during the project have sought to address these diverse requirements, identifying and consolidating those aspects which have implications for the healthcare record itself.

The Requirements for Portability Deliverable explored the problems accompanying the current diversity of computers, networks, data formats, communication protocols and application programming environments. The work in progress within the standardisation bodies was reviewed, and the implications this has on the portability of the healthcare record. The decision was taken to adopt a formal object modelling approach to express the GEHR architecture in order to overcome many of these problems.

The Deliverable included an extensive investigation of the language translation aspects of portability, and the recommendations from this work have been incorporated into the GEHR Architecture.

The Requirements for Communication Capacity Deliverable reviewed the evolving standardisation work in the field of data storage, manipulation and telecommunication. Much of this work has been reflected in the proposals for a GEHR Exchange Format.

The requirements for the storage and transmission of different data types were explored in detail, including clinical data sets, laboratory, drug prescription, radiology and biosignal data. In many cases the standards are still evolving, and in order to cater for future developments and for new data types GEHR has adopted a generic approach to the handling of bulky data. The object model requires that bulky data types are held along with a method by which that data can be accessed and communicated.

The clinical requirements were drawn together into a clinical functional specification for healthcare record systems. This Deliverable included proposals for the structural organisation of healthcare record entries and for safeguards to preserve clinical meaning and accountability.

In order to provide a rationale behind the main architectural constructs, the individual clinical Deliverables are summarised in more detail below.

3.2 Summary of the Requirements for Clinical Comprehensiveness

3.2.1 The Context of Clinical Records

This Deliverable (consisting of 144 pages with 124 references) lays out the set of clinical requirements which must be met by a computerised patient record if it is to be comprehensive, communicating and portable whilst supportive of a high standard of clinical care. These rest on the results of the first-year workplan of the GEHR Project: the requirements for clinical comprehensiveness in primary and secondary care, across specialities and data types, and the testing of prototypes in these settings. Further requirements identified here are based on previous work, liaison with other AIM projects and national initiatives in the field.

It is important that the set of requirements embracing such a large field of reference be assigned relative priorities so that any compromises which prove necessary when specifying the architecture are consistent with the real world in which the record will be used. The priorities of the GEHR Project reflect the belief that the healthcare record is most necessary, and should be most available, when a clinical member of staff is offering care or recording the care they have given in a consultation. Thus compromise should always be directed towards offering the most relevant information to an individual carer when attending a patient.

The recording of healthcare information is selective and will always involve a compromise, largely because of time and space constraints. We need to be aware of the danger that the growing enthusiasm for a better organised and more statistically useful document may presume too greatly on the human factors involved in creating it.

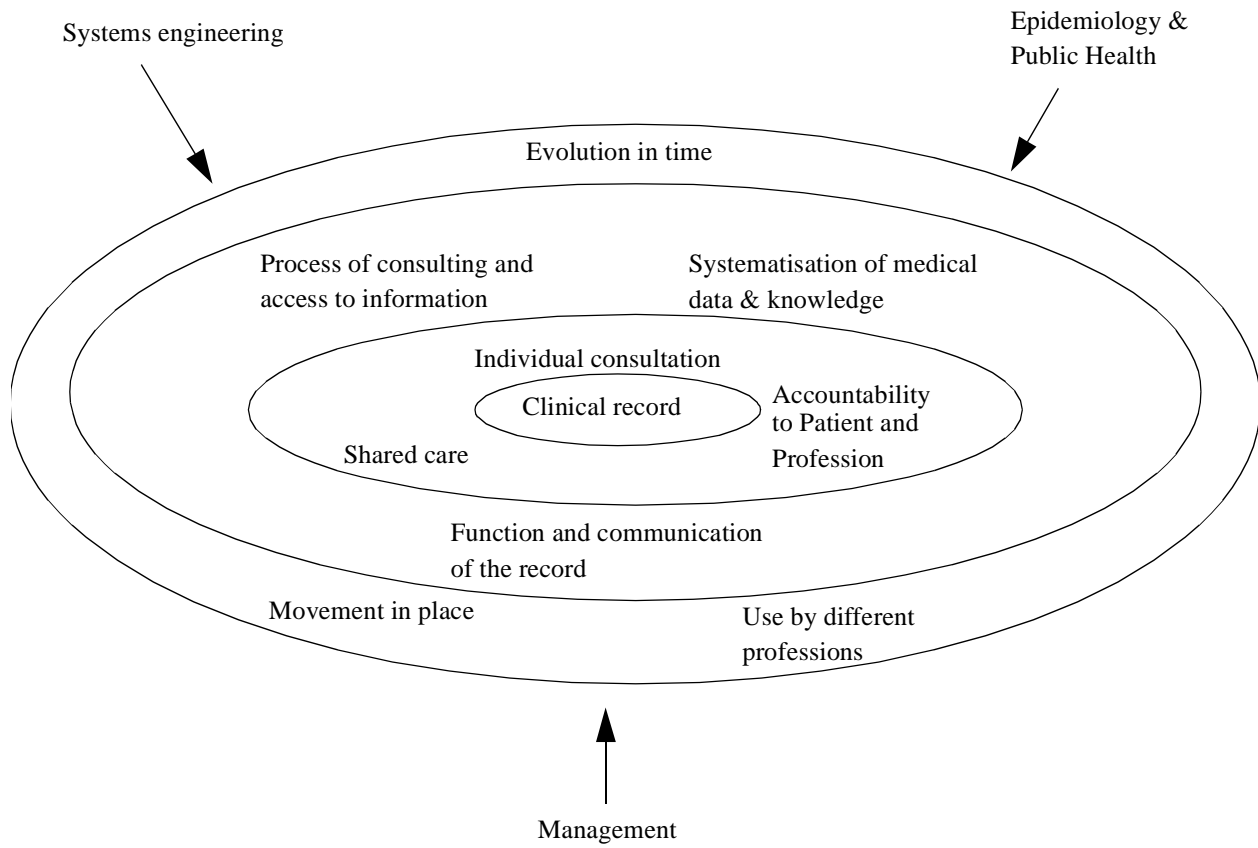


FIGURE 2 The Context of the Clinical Record

The clinical record will be used by staff trained in different disciplines, working in different settings, on different sites, and in different languages. The architecture must facilitate record storage on different sites and provide a common interchange format between heterogeneous systems.

The record architecture must accommodate the current growth towards the systematisation of medical knowledge. This involves issues of terminology, classification and a fundamental understanding of the basic sciences of medicine and their clinical correlates. Given the use of the record in the individual clinician/patient consultation, it is clear that it must be ordered around a realistic support of the processes of clinical care and the requirements for access to information. It must take account of the wider needs for communication of the record which must traverse all aspects of the health care services, and cross regional and national boundaries.

The healthcare record must accept three areas of change; in time, place and clinical perspective. A record evolves gradually over a person's lifetime, and family records over generations. We know that people's health care needs change and evolve in time, as does the practice of medicine, and the economic and social framework within which medicine is practised.

It is recognised that a computerised healthcare record will be developed within the context of

current technology and systems. The record has to be responsive to public needs and priorities, which go beyond personal health care, and which must be debated within an epidemiological and public health context. The evolution of solutions will require work in all areas covered in the diagram (Figure 2), and within an over-view of the whole, which recognises the strengths and limitations of what has been achieved, and the opportunities which arise from new developments.

3.2.2 The Historical Background of Clinical Records

Some of the oldest surviving examples of medical recording are papyri from ancient Egypt which contain details of surgery and prescriptions. There has always been a recognised need for those involved in healing to pass on details of successful procedures or potions either by written methods or through an oral tradition. Throughout this century, institutions have attempted to influence the data collected within their walls but there has always been resistance to standardisation on the grounds that the freedom of individual clinicians must be protected. A 1923 textbook noted that “From the standpoint of scientific record taking, case histories are most glaringly defective in what they fail to record about a patient¹.”

It could be argued that if pro-forma had been widely introduced in 1923 they might have hindered the development of new ideas in medicine by discouraging new observations and thoughts. In 1957 Balint published “The doctor, his patient and the illness” which recognised the psychological basis of many health problems². This book has had a huge impact on medical practice and therefore on the content of medical notes which now tend to contain much information relevant to an individual's psychological well-being such as sources of stress, social interactions and perceptions of illness.

In 1969 Weed published “Medical records, medical education and patient care” which introduced a method of structuring a record, the Problem Orientated Medical Record (POMR)³. This was a format for clinical recording consisting of a problem list, a data base (that is, the history, physical examination and laboratory findings), and then, written out separately for each problem, a plan (diagnostic, therapeutic and educational) and a daily SOAP (subjective, objective, assessment and plan) progress note. However the POMR was not widely adopted exactly as Weed proposed because it proved to be too time consuming.

As both primary care and secondary care organisations have become more complex, instead of the notes just acting as an aide memoire for an individual clinician they have become important as a means of communication between clinicians. The involvement of different professionals in care and the recognition of the inter-relationship of physical, psychological and social factors has led to notes becoming vast repositories of data with little structure to facilitate the processing of this data.⁴

-
1. Pearl R. 1923 “An introduction to medical biometry and statistics” W.B. Saunders Co, Philadelphia. (p91)
 2. Balint M. 1957 “The doctor his patient and the illness” Pitman medical publishing, London.
 3. Weed L. 1971 “Medical records, medical education and patient care: the problem orientated record as a basic tool” The Press of Case Western University, Cleveland, Ohio
 4. Gregson B, et al. 1991. Interprofessional collaboration in primary health care organisations. Occasional paper 52. RGGP, UK.

As the process of human reasoning has become better understood, it has become apparent that logical thinking coexists with less well understood “intuitive” processes; both modes of thinking and reasoning are important in medical decision making¹. This insight into cognitive processes has led to the current situation where there is an apparent paradox between the need for more “structured data” which is used in a logical way to derive a conclusion and the need for “narrative data”. This need to value and listen to the story as told by the patient is explored in “Doctors Stories” by Kathryn Montgomery Hunter². She suggests that when physicians have a working knowledge of life histories and a sense of medical narrative that can accommodate the experience of illness, they are better able to provide good medical care, especially for those they cannot quickly cure. This tension may only be an apparent paradox as both types of data are important and can coexist. A structured approach will ensure all the necessary information is acquired to arrive at the right decision. However within any structure there must always be the facility to record events as described by the patient (not in a processed form) and the ability to record data that does not immediately seem significant.

3.2.3 Clinical Competence and the Clinical Record

Clinical competence is at the heart of all medical practice; a fundamental requirement for the healthcare record is therefore that it should enable and reflect the competence of the clinician who creates it. The specific components of competence include:

the consistent ability to select and perform tasks employing intellectual, psycho-motor and interpersonal skills:

- to deliver curative and rehabilitative care
- to promote health
- to organise preventive activities
- to plan, organise and evaluate health education activities
- to collaborate with other agents of community development
- to participate in research
- to manage his or her services /resources
- to train other members of the health care team
- to participate in and sometimes to lead the health care team
- to engage in self directed learning
- to engage in self evaluation and quality assurance

the consistent demonstration of appropriate moral and personality attributes:

- honesty
- self awareness

-
1. Dreyfuss H, Dreyfuss S, Athanafion T 1986 “Mind over machine (the power of human intuition and expertise in the area of the computer)” Blackwell scientific publications, Oxford.
 2. Montgomery Hunter K. 1991 “Doctors Stories: The narrative structure of medical knowledge” Princeton University press, Princeton, New Jersey.

- empathy
- respect for patient autonomy
- confidentiality.

How can a healthcare record promote and support honesty in the clinician? Clinicians need the ability to modify the record by explaining, reinterpreting and commenting on their actions without altering their initial record. Patient autonomy is promoted by ensuring that as far as possible patients are informed. In particular the record must be accessible to patients and avoid unnecessary jargon. Confidentiality and the question of who has access to the record must be defined. Perhaps the patient should decide the level of access to sensitive information that is afforded to any given health professional.

Some components of clinical competence are closely related to the role of clinicians in the societies in which they practice. The healthcare record must not impose the values of one society on the clinical practice of another, although it should promote ways of learning about different styles of clinical practice. The healthcare record must be capable of evolution as society develops and defines some aspects of the common core of practice. The record must be dynamic, a true contemporary record of each encounter which maps directly onto a different context as changes within society and the profession are translated into legislation.

The medical profession and the public have a right to expect that when a competent clinician uses The Good European Health Record, its structure will allow a fair assessment of patient care in its widest sense.

3.2.4 The Role of the Clinical Record

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 structure for the roles fulfilled by the record is based on a list originally proposed by Shortliffe & Barnett.

- 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 health care purchasers (health authorities or insurers) for quality assurance
- health care 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!)

3.2.5 The Nature of Clinical Data

Complexity

Apparently simple elements of healthcare information can at times require quite complex recording structures. In the example of blood pressure measurement, for many clinical purposes a combined entry for systolic and diastolic readings (e.g. 120/80) will be adequate. For monitoring the progress of a patient's results graphical representation will be helpful, but would necessitate the separation of the readings into systolic and diastolic pressures. At times a blood pressure entry may have the value "Unrecordable" or a combination of a numeric entry and a text entry:

"Systolic: 60, Diastolic: unrecordable".

Blood pressure may be measured in different positions (lying, sitting or standing), at different sites (e.g. arm or leg) and by different methods (sphygmomanometer, intra-arterial) which may themselves be further specified (e.g. large cuff). Narrative comments relating to the patient's state of anxiety during the consultation may be important. Specialist centres and research projects may require the documentation of extra information which cannot be predicted in advance.

All of these additional attributes of blood pressure might have clinical significance and need to be documented in the record. They might on occasions have a direct bearing on the way that a single blood-pressure reading is interpreted, and the GEHR architecture must cater for these possibilities. Much of healthcare data can at times require complex recording structures, which may vary depending upon the institution and with time.

Certainty and precision

Data of all types when recorded carry with them degrees of uncertainty. This relates to all information, but especially to clinical findings and interpretations. Use of language in describing uncertainties is often ambiguous, and Bryant and Norman ¹ have demonstrated the wide disagreement about the meaning of common terms, such as probably.

The assessment of severity or risk in a situation is often as important as the recording of findings and may be the sole basis for management decisions. Moves towards the progressive computerisation of the healthcare record will require a recognition of these aspects of the clinical information to be expressed.

Diversity of Data Types

The healthcare record is made up of a collection of entries, which are at present usually written in paper records, frequently in the form of narrative data. Within this there are many short hand abbreviations such as “SOB” (shortness of breath). Some terms such as “acute myocardial infarction” have become standardised in their usage and are increasingly being drawn from term sets. Many medical data have numeric values (e.g. weight, serum cholesterol), and it is particularly in such data that the notion of precision becomes important.

Clinicians frequently make clinical drawings to describe abnormalities detected -this is often the most concise way to make such recordings. Coupled with this is the use of symbolic diagrams to convey concepts. Some specialities, such as Ophthalmology, have highly stylised symbols and drawings which are for communication within the speciality. ECGs and EEGs are expressed as analogue data, usually stored graphically, and many test results are now in image form (X-rays, ultrasound etc.).

The range of such methods for conveying information is not static and will evolve as medicine itself progresses. The GEHR architecture must provide for the full range of multi-media data types to be incorporated within the record. The tools for recording and analysing these forms of data are an equally important part of this process and must be catered for.

3.2.6 The Structure of Medical Language

There are many classification systems used in medicine, and a shared healthcare record must allow use of any one or all of these systems. They are usually designed for a specific purpose and may be used in specific or general settings. The problems with all of these systems are readily apparent:

- each classification is only primarily valuable to the specialist group that designed it
- even users within that group can find it too rigid or need considerable training
- the limited vocabulary constrains natural expressivity and may artificially skew information

1. Bryant GD, Norman GR. (1980) New England Journal of Medicine, 302:411

- it may be difficult to link terms together meaningfully
- there may be insufficient room within the code set for the future expansion of medical knowledge
- the concepts behind the structure eventually become outdated

The development of an electronic healthcare record architecture needs to support the recording and communication of data derived from term sets, but avoiding the pitfalls of many current classification systems.

3.2.7 Evolving From Paper Records

During the first year of the project many groups of clinicians were consulted to elicit their attitudes to the growing trend towards a paperless consultation. As this process of evolution continued, a number of current features were felt important to be retained. Many of these are applicable to healthcare records in general, irrespective of whether they are on paper or computer. It is not always easy to separate the features of a healthcare record architecture from those of a computer system based on it, but in this document an attempt has been made to focus on those requirements which could usefully inform the development of the architecture.

Locating the record

Paper medical records are not always readily available at the point of clinical care, particularly in large institutions. A computer-based record has the potential to overcome this problem, but care must be taken to ensure that access to a computer is not inadvertently also made difficult. Clinicians will, for example, need to take a useable record with them on home visits.

Locating information within the record

Clinicians almost invariably are working in consultation under a pressure of time with sometimes only one or two minutes allocated to updating the record. They have acquired a considerable skill in assimilating salient points from a medical folder which often is in considerable disorder, and often without summaries. Navigation through a computer record should aim to be as intuitive as possible.

There is a growing trend for analytic tools to be available to users during the process of data entry on computer, including classifications and coding systems. Finding the balance between structuring data entry through templates, and leaving enough room for individualities and fine detail, is one of the challenges to be addressed.

Comprehensiveness

The healthcare record must contain or reference all information thought to be clinically relevant to the care of a patient. The ability to view information at various levels of detail will also mean that information can be retained in the record without overwhelming the clinician providing care. For example, a blood test result may need to be viewed in great detail on its initial return, though may be scanned as “normal” in future, though its individual values may be accessed for purposes of a graphic display for assessing trends.

Expressiveness

While at times, or in some locations there may be an overwhelming need to have a healthcare record with rigid protocols, decision support and management plans, other clinicians will require expressive narrative to convey their findings.

Faithfulness

Rector and colleagues identify three aspects of faithfulness¹;

- Faithfulness to the clinicians observations of the patient
- Faithfulness to the decision making process
- Faithfulness to the clinical dialogue.

The healthcare record should ideally be structured in a way that preserves the meaning of the information when it was originally written, so that it can be understood if read by another person elsewhere. Where the information has been re-presented, for example by language translation, that the reader should be made aware of this and have access to the original entry.

Sharing of medical information

A major perceived benefit of computerisation is the ability to have many different views of the same data, allowing different professionals use of the same information, and reducing the need to enter similar data many times in the record.

Adaptability

The practice of medicine is evolving in many areas at a rapid rate. Attitudes to record keeping can change dramatically in a short time, and innovations may lead to totally new data being recorded in specialised centres. Doctors and other health care workers get pleasure from tailoring their notes to their own needs and interests. Adaptability is a major requirement of any paper or computerised healthcare record.

Author Responsibility

It is widely recognised that every entry in the record must be attributed to an author. What is displayed in the record should be a true account of events as recorded by the author. If changes need to be made, it must be possible to step through and account for those changes individually within a rigorous audit trail.

Security of the record

Accessing and reading paper records lying on an office table or even stored in a central store-room is far from prevented by current institutional security measures. Any computer record which is even more widely accessible on a network will need to have a greater degree of security protecting access. A very sophisticated audit trail will need to be kept, and it may be that the record will be partitioned to tailor access to individual rights.

1. Rector A, Nowlan W, Kay S. 1991 Methods of Information in Medicine 30:p176-86

3.2.8 The Importance of Data Security

It is important to emphasise that security must be a major requirement and is indeed the foremost concern in most patients minds when the subject of computerised patient records arises. Computerised patient record systems have two requirements. First patient and provider privacy must be protected. Second, data and software must be safeguarded against tampering and unintentional destruction. These requirements demand both system and data security measures.

- There must be a watertight method to identify the author of the record (electronic signature).
- It must be possible to update the record but it must be impossible to alter or erase previous entries completely.
- If the record is to be used by a large number of professionals for different purposes it must be possible to withhold certain information from general viewing.
- Electronic health records must be secured against illegitimate use.
- There should be an agreed set of information recorded with every entry, which might include: time and date, identification of provider (personal ID, name, position, level of competence, physical location, telematic address), identification of coding system used, definition of ownership of the information and who is permitted to view it.

3.2.9 Drug Prescription and Its Incorporation Within the Clinical Record

The ability to store drug information in the patient record is a key requirement for the GEHR Project. In addition, the prescription is not an isolated communication of instructions between the doctor and the pharmacist who is expected to dispense the drug, or others directly involved with the care of the patient. It increasingly involves health administrators, epidemiologists and other medical professionals who wish to have access to the anonymised information for their own purpose.

The ability to interchange drug data must also be a feature of the Good European Health Record.

3.2.10 The Role of the Clinical Record in Education

The work in this area has been published in a separate Deliverable, described later. A summary of the key requirements is listed below.

- Computerised records should support developments in health-professional education, in particular self-directed and problem based learning.
- Computerised healthcare records should be accessible to students at an early stage in their education. The user interface should be designed so as to make it easy for the inexperienced user to find his/her way around the system.
- Adequate safeguards need to be established to ensure privacy, confidentiality, and data protection. For instance, there may need to be methods of stripping records of personal identifiers.

- Computerised records should be viewed as a rich resource for analysis and study by students. Clinicians, epidemiologists and medical educators will need to design activities which encourage students to use records as a learning resource. There will need to be easy access to statistical support tools.
- The relationship between data quality and patient care should be obvious, encouraging students to take responsibility for patient data.
- It must be feasible to directly access decision support tools and bibliographic databases.
- Computerised healthcare records should interact with educational software so the latter is seen as relevant and supportive of good clinical care.

3.2.11 Summary of Research Investigations Undertaken

A Study of Paper Records in use at St. Bartholomew's Hospital

The St. Bartholomew's group undertook two reviews of current paper hospital healthcare records. The first survey was to look at the structure of medical and nursing notes and at the types of headings and subheadings used. The second survey was carried out in association with the hospital's department of medical audit and compared the notes made by different health professionals to look for common content and common processes.

Conclusions

- Each profession does, within a certain shared process, have a unique approach and different needs.
- There are many synonyms in use in doctors notes and the availability of these is necessary to make any system acceptable to individual users.
- Headings, subheadings and contents need to be interchangeable.
- Clinical notes need to be able to express uncertainty.

A Study of the Use of Computerised Health Records

The Clinical Task Group has carried out a Europe-wide survey of the current suppliers of computer software offering a healthcare record system. It is clear that in each country a diversity of systems exists, and that it is rare for any one system to radically outstrip any of its local competitors. This situation would suggest that real excellence has not yet been achieved by the computer industry in the field of healthcare recording. Anecdotal experience confirms that many of these systems are either very comprehensive but clumsy to use, or are user-friendly at the price of comprehensiveness.

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. It would therefore be difficult to envisage the transfer of clinical information between these systems without a very complex mapping process which would be vulnerable to any future changes in the systems.

A Study of the Current Health Record Architecture

An important strength of the project methodology has been the pre-existence of a multilingual prototype exhibiting some of the features which will be required of the record architecture. The work of the project has placed a strong emphasis on the evaluation of successive generations of such prototypes in field test sites. The prototype tested has been referred to as the Current Health Record Architecture.

The early evaluations were performed on the initial prototype application which had already been in use within Belgian General Practice and French Hospital Practice. The studies consisted of the analysis of the text content of healthcare records created over the preceding year, and questionnaire surveys of experienced and novice users of the EHCR systems. The prototype was also customised to meet the requirements of a UK research project in General Practice focusing on the implementation of clinical guidelines in asthma and diabetes; the applicability of the architecture to guidelines use was therefore assessed.

Conclusions

- Computerised records must integrate a large number of terms and their number and categories must be expanded in an evolving and coherent way. They must also offer structures that allow for the various uses and combinations which reflect the many approaches and specialities in health care. Computerised records should support protocols. Some clinicians will wish to use a structured format for their consultations, others will require a much more open format for their consultations.
- There should be no loss of word accuracy if an electronic record is used instead of pen and paper for textual data, and there will need to be alternative methods for data entry to the conventional keyboard.
- It should be possible to include laboratory data, ECG recordings, images and drawings in the electronic record. There must be logical links between a medical request (test) and its answer (result), and it must be possible to vet all results of investigations before they are entered into the electronic record.
- Clinical drawings should be incorporated into the data structure in such a way that the information they contain can be submitted to analyses comparable to those allowed for numeric or coded values.
- There must be sophisticated methods of statistical analyses both within an individual record and across populations. The ability to analyse data at the time the patient is consulting is a useful and popular function.
- The ability to automatically generate commonly required documents is a popular function, and the electronic record should speed up tasks of administration and data entry.
- Each professional group looking after a patient will wish to have a “view” of the patient's record which enables them to read subsections of the notes in a meaningful way. However they will at times wish to consult notes made by another

professional group.

- There must be communications and links with other information systems
- Patient data must never be lost.

An Investigation of the Range of Medical Vocabulary in Use

An analysis was performed of the medical words used in over 20,000 referral letters, from within General Practice and from Hospital Departments in the UK and Luxembourg. The letters were all written by dictation and taken from recent chronological archives. The overall frequency of occurrence of individual words was counted using a computer programme. The frequency tables provided an overview of the clinical vocabulary in common usage, and has supported the later development of the GEHR Term Set.

Conclusions

- Clinicians must continue to be able to use a rich and varied vocabulary.
- Term-browsing facilities must be provided in a user-friendly manner to minimise the risk of spelling and typing errors.
- It should be recognised that proper nouns and synonyms will represent a substantial proportion of the medical text.

An Investigation of the Use of Drawings in the Clinical Record

Clinical notes often contain sketches and diagrams done by clinicians. They are included either because they are a quicker way of recording information or a clearer way of recording information. They are particularly useful when recording the location of something. It is quite hard to accurately describe the location and extent of something like a jagged wound in words and so a sketch is used instead. Many people like to explain things to patients using sketches.

A questionnaire survey was performed amongst doctors in the UK and France to determine the extent to which clinical drawings were considered an important component of healthcare records, and to compare the consistency with which a given set of clinical findings would be annotated.

Conclusions

- Drawings are commonly used to depict clinical information
- There is a wide variety of drawings in use
- Drawings made by different clinicians are fairly standard in their style and the way information is depicted
- Drawings are seen as a “quick” way of recording information and any computerised method must be quick and easy to use
- Drawings are often used in communication between clinicians and patients
- Drawings are sometimes the only record of a Transaction between a clinician and a patient and must therefore be stored securely and be transferable

An Appraisal of the Incorporation of Images in the Clinical Record

Surveys among French doctors and UK doctors have shown that both groups wish to see X-rays, ECGs and parts of other biological recordings as appropriate. Both the French and UK doctors wished to have the benefit of a specialist's report when viewing images or complex biosignals. There was great interest in viewing as many of the investigations done as possible. Most doctors were most interested in X-rays and ECGs but felt that access to edited parts of more specialised investigations such as EMG and EEG would educate them and increase their job satisfaction.

Both UK hospital doctors and general practitioners agreed that no investigation should ever be entered directly into patients notes. All results of investigations should arrive at some designated point ("Mailbox") and be checked by the responsible clinician before being entered into the patients notes.

3.2.12 Overall Conclusions

There are clear advantages to the clinicians themselves of well-organised electronic healthcare records for improving the completeness of the information they elicit from the patient¹, the scope for the subsequent analysis of that data², for supporting shared care between clinicians^{3 4}, and to demonstrate clinical competence^{5 6}. Improving the ease with which EHCR applications can be learned and used⁷ and developing more efficient means for clinical data entry⁸ have both been suggested as solutions. However, the inherent diversity and complexity of medical data⁹ and the need to use rich and varied descriptive terms^{10 11 12} are held by many clinicians to be a fundamental obstacle to adopting more formal recording structures. The lack of an appropriate architecture for healthcare records has been identified as a major impediment to progress in this area¹³.

-
1. Lilford R, Kelly M, Baines A. Effect of using protocols on medical care: randomised trials of three methods of taking an antenatal history. *BMJ* 1992;305:1181-1183.
 2. Spann SJ. *Journal of Family Practice*. 1990;30:457-464.
 3. Shortliffe EH, Barnett GO. *Medical Data: Their Acquisition, Storage and Use*. *Medical Informatics: Computer Applications in Health Care*, Chapter 2. Addison Wesley Publishing, Reading, Massachusetts. 1990.
 4. Pagano MP. *Communicating Effectively in Medical Records*. Sage Publications, London. 1991.
 5. *Proposals for New Performance Procedures: a Consultation Paper*. General Medical Council, London. 1992.
 6. Southgate L, Berkson L, Fabb W et al. *Towards Better Definitions of Competence: a Paper From the Cambridge Conference*. Office of the Regius Professor of Medicine. Cambridge University. 1989.
 7. Skifjeld K, Harket G, Hurlen P, Piene J, Skjervold S. *A Document Architecture for Health Care Records Integrating Structural and Semantic Aspects*. AIM Office. 1992.
 8. Rodnick J. *Journal of Family Practice*. 1990;30:460-464.
 9. Blois M. *Information and Medicine: The Nature of Medical Descriptions*. Berkley and Los Angeles: University of California Press. 1984.
 10. Arnold R, Povar G, Howell J. The Humanities, Humanistic Behaviour, and the Humane Physician: a Cautionary Note. *Annal. Int. Med.* 1987;106:313-318.
 11. Bryant GD, Norman GR. *New England Journal of Medicine*. 1980;302:411.
 12. Hunter K. *Doctors' Stories: The Narrative Structure of Medical Knowledge*. Princeton University Press, New Jersey.
 13. Rector A, Nowlan W, Kay S. *Methods of Informatics in Medicine*. 1991;30:179-186.

3.3 Summary of the Ethical and Legal Requirements

In this Deliverable (consisting of 68 pages with 52 references) the ethical issues raised by the application of information technology to the health care record are addressed. The term 'healthcare record' is used throughout as it does not imply that the use of the record is confined to doctors. It does however imply that the discussion is confined to that part of the record created by clinicians, with specific reference to administrative information where relevant. In addition, the boundary of the record has been defined as all recordings made by a responsible clinician regarding the care of the patient. Thus information does not form part of the EHCR until a clinician has taken responsibility for that information and committed it to the record.

Ethical issues are fundamentally important because in developing an electronic health care record (EHCR) there is a risk of serious harm to patients or clinicians which involves the EHCR and its processes. However the risk can be minimised without compromising the usefulness of the record and regulation is both technically feasible and morally appropriate.

As many of the ethical issues arise from the purposes of the EHCR these have been made explicit, and assigned to a hierarchy which will itself aid the resolution between competing ethical imperatives.

- The primary purpose of the EHCR is to benefit the patient by providing a record of care which supports present and future care by the same or other clinicians.
- The secondary purpose is to provide a medico-legal record of the care provided and hence support and demonstrate the competence of clinicians.

Tertiary purposes must be legitimate (involve consent) and can never be allowed to compromise the primary or secondary purpose. Examples of tertiary purposes are the generation of data for health service management or public health programmes.

The document addresses these issues, the dangers of the abuse of technology, and the moral and legal principles behind a 'Good' Electronic Health Care Record. The foundations of the relationship between a clinician and a patient are the delivery of clinical care to the highest standard and the respect for patient autonomy. This inevitably leads to the conclusion that the right to informed consent and the right to confidentiality are also moral principles of the highest importance behind a 'good' EHCR. Patients should exercise as much choice over the content and movement of their healthcare records as is consistent with good clinical care and with the lack of serious harm to others. Records should be created, processed and managed in ways that optimally guarantee the confidentiality of their contents and the legitimate control of patients over them. The record must be secure yet accessible to patients.

The document also discusses the legal principles which have a bearing on an EHCR in terms of:

- confidentiality;
- ownership and copyright;
- liability and accountability;
- identification;

- durability;
- processing of personal data;
- transparency.

The present legal diversity is the result of uncoordinated and piecemeal legislation and there is a need to harmonise legislation if movement of medical records is to be sanctioned by clinicians and patients.

Consideration of these moral and legal principles and the dangers in creating an EHCR led to the definition of several major areas which must be addressed and which are listed below. Each one forms the focus of a chapter in the document.

1. The limits of patients' control of the creation, movement and processing of the health record.
2. The limits of the control of access to the contents of the health care record by patients and clinicians and others.
3. The establishment of individual accountability through the EHCR as a physical record of the contact between clinician and patient, and to avert potential negligent use of the health record.
4. The appropriate patterns of security enforcement and organisation. Audit thereof for protection of individual privacy, including professional guidelines for appropriate 'whistle blowing'.
5. The creation of educational processes which inform both patients and health professionals about their rights and duties.
6. The role of regulation in the development of the EHCR.
7. An analysis of what forms of regulation, legislation and processing may be logically lead to increased risks to patients and clinicians.

3.4 Summary of the Educational Requirements

3.4.1 Introduction

This Deliverable (consisting of 62 pages with 35 references) was primarily the result of a collaboration between members of the GEHR consortium based in the Medical College of St. Bartholomew's Hospital, London, UK and members from the University of Oporto, Portugal. The initial sections contain the results of an extensive literature review and a survey of medical schools. The final section is a conceptual model which was carefully constructed to meet the requirements described in the first two sections of the document.

3.4.2 Preparing Healthcare Professionals to Manage Patient Information

Modern healthcare is dominated by the need to integrate and process information. An episode of illness generates information from the patient, from investigations, and from different professionals. This information needs to be integrated with what has previously happened to the patient, and with relevant information about the disease and treatment. Most healthcare professionals feel apprehensive about their ability to assimilate and process the information they encounter daily. These concerns have been taken up by innovators in medical education. New approaches to medical education have been proposed to help with these problems. The basic premise is to move away from healthcare professionals having to remember vast amounts of information, to a situation where healthcare professionals are able to identify problems, work out what knowledge they require to deal with those problems, and then to seek out that information. It is argued that this process is dependent on a more organised approach to the medical record. Unfortunately healthcare students do not currently receive much training in the use of healthcare records. A survey of medical students showed that they are confused about variations in how notes are written, and the discrepancy in how they are taught to write notes and the notes written by qualified doctors. They also perceive that they need to be computer literate, but at present do not think they are getting adequate tuition.

3.4.3 How Electronic Records could Benefit Education

It is argued that electronic records could encourage students to develop a more positive interest in the process of recording patient data. If they learn to be systematic and code data, they will be able to use the medical record to access clinical decision support systems and for bibliographic and knowledge links. This will enable them to become more independent in their learning. If they have access to databases of medical records, they will be able to learn about diseases by comparing and contrasting similar cases. In the Beth Israel Hospital Boston, students are able to use computer terminals to view patient test results. Students at the University of Kansas medical school are loaned portable notebook computers which they use to access computerised medical record systems at four different community hospitals. This enables them to follow up patients they have encountered. Several new activities are proposed which use electronic records to help students learn about recording data and patient care.

3.4.4 Requirements for the GEHR Architecture to Support the Use of Electronic Health Records in an Educational Environment

It is important to understand the current ethical framework which is observed when healthcare students learn about patients. Two possible scenarios are discussed in the document. In the first scenario, a student seeks permission from a patient to be involved in his or her care. The student has established a relationship with the patient and therefore has permission to read that patient's notes, and may sometimes enter data in those notes. In the second scenario, a student is observing a clinician with a patient. The student would not normally make entries in the notes in this circumstance, but the clinician or the patient may give the student permission to read them. It is proposed that these two scenarios should be reflected in the way students access electronic

healthcare records. Students should not look at an EHCR outside these contexts unless they are looking at anonymous data.

As all healthcare professionals learn by apprenticeship, it is of fundamental importance that students feel actively involved in patient care, and responsible, by making some of their observations in the 'real' notes. To ensure that students realise the importance of accurate and conscientious data collection, it is proposed that they have access to a database of anonymised cases which can be used for learning activities. In order to achieve a workable, ethically acceptable system we have proposed three concepts:

1. Shadow notes
2. Dummy system
3. Logging facilities

Shadow notes

Shadow notes are a method to ensure students can experience using the 'live' medical record system safely. Students are given access rights to the electronic medical record system. They are clearly identified by the system as being students. They view the same medical record as a qualified professional and enter their observations as a qualified professional would. However, the system identifies these records as having 'student status'. This is done so that it is clearly visible to subsequent viewers of the record that the observations were recorded by a student. Student notes have different properties to other entries: they are excluded from analyses and are not transmitted to external institutions. It should be possible for a qualified professional to validate a student's entry, document that they agree with the student's notes and change the status of the student's notes to that of qualified professional. The student's notes would then have the same appearance as the 'live' record, be included in analyses and be transmitted to external institutions.

Dummy System

The rationale for this, is to provide a place where students can practice using the technology without disrupting the 'live' system. The dummy system would also provide a large database of cases so that students could practice manipulating aggregated data and learn from analysing large numbers of cases. The dummy system would have the same functions as the live system. Cases could be copied from the live system, but should undergo an automatic anonymising process. Patients should give permission for their anonymised data to be used for teaching purposes.

Logging facilities

It would be very helpful to be able to identify patients followed by particular students during their training. Each student could present a collection of such cases to a tutor to form the basis of a tutorial. Students could be provided with a report of what type of patients they had seen during their training (this would not identify patients). This would be useful for personal audit but could also be used for institutional audit to check students were gaining a wide enough experience.

3.5 Summary of the Clinical Functional Specification

3.5.1 Introduction

This draws together the work of the Clinical Task Group of the GEHR project. It proposes a specification for the GEHR architecture, based on the earlier requirements Deliverables, and expressed largely in clinical terms. The originating Deliverables have all been extensively referenced to the published literature, but in view of the more derived nature of this document the statements here have not been individually referenced.

3.5.2 The Transaction

Some people have adopted a global view of the healthcare record and 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. 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 it has been authenticated by a responsible clinician. 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 which protects both patients and clinicians.

This Specification proposes that there should be a clear border to the electronic healthcare record. This is formalised through the concept of a *Transaction*. The Transaction will commonly be recognised as a patient contact or a consultation, but may at times reflect an interaction with the record when the patient is not present, such as filing a test result or a letter. These Transactions are authored by clinicians who accept responsibility for the accuracy of the information added during a Transaction, and they encapsulate the cohort of information which has been entered as one 'interactive session' with the record. All clinical data within an electronic healthcare record should exist only within such Transactions. All Transactions should be available for medico-legal scrutiny, even if they have been subsequently amended.

In order that the electronic medical record may grow logically and in a way that preserves its integrity, the Transaction forms the basic unit of the clinical record. This notion applies to the record, not to our clinical practice; so a single consultation with the patient may lead to one or more Transactions.

The principle of a single logical record is an essential ingredient for a shareable and portable clinical record. This means that if there are multiple health records, they can be combined at any time to provide a single record of care.

Medico-legally, it is essential that there can only be one recording of a particular patient consultation, by a particular clinician at any one moment, although this recording may be subsequently corrected or amended. By defining Transactions uniquely using these criteria, the logical integration of two record sources will always safely maintain each Transaction within its context in the combined record. When patients are being cared for at two sites, such as a hospital and general practice, the record may either exist as a single entity shared by these two sites, or individual Transactions may be transmitted to allow the record to develop independently on each site but with some shared information.

The healthcare information within an EHCR must exist solely within Transactions, each uniquely identified by the time, the institution and the clinician responsible for its creation. For medico-legal reasons it must also be possible to determine the date and time at which Transactions have been added to an EHCR, whether created at that institution or received from another EHCR source. This might, for example, allow an institution to demonstrate that historic information of clinical significance had been received after the recording of a consultation with a patient.

3.5.3 Transaction Types

Each Transaction relates to a single patient, a single responsible person, a single date and time, and a single Transaction type. We propose a limited number of Transaction types which can be shared by all users to enable the record to function predictably.

Administration

This Transaction type plays a crucial role in identifying the patient. It will be used to record any information which assists in the management of a patient but which is not specifically related to their health status e.g. name and address.

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. It may often be necessary to define further the type of consultation, the location of the consultation, the type of clinic, or other specific information relating to it.

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 within a summary Transaction. This corresponds to the current concepts of clinical summary in paper records, but could be extended for summarising any type of life or health event.

Trigger

Any condition or information requiring action at a future date, or circumstance. It may require mandatory elements such as the date and time for this information to be brought to the carers attention, the date and time the action falls due, the relationship between the information and the date (e.g. no later than, no earlier than etc.). In many instances, a trigger Transaction will resemble

setting an alarm clock to broadcast a message on some future occasion.

The trigger must be capable of being invoked by a variety of stimuli, including dates, times, and clinical events. The action triggered may be a prompt to the user, or a protocol. Prompts may be set to have semi-automatic capability, for instance generating a recall letter without prompting - however they must never be capable of automatically amending the record, such as a repeat medication drug dose if a new weight is entered.

Report

These Transactions are used for information which has a legal status outside the record. Thus report Transactions involve communication from one responsible person to another. Clinical letters, information from a third party, requests for and results of tests, and recording an opinion or proposed action (when the patient was not present) would be examples of this. If any information from parts of several Transactions are compiled into one new assembly, such as a table of results, these could also be stored in the record as a report.

Continuing 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. In some ways this will resemble a summary Transaction, except relating to the future rather than the past. Care plans and current or repeat prescriptions are examples of this.

Nota Bene

This Transaction type will be defined by its behaviour, as the information will be displayed whenever the record is opened. It is thus critical information relating to this patient, which the last clinician requires the next clinician to see. In many ways it can be seen to resemble the outside cover of the notes folder on paper - which is intended to be noted by all who access it, and may include major allergy or disease warnings.

Note:

The concept of episodes of care has acquired a growing importance, often for resource and managerial purposes. However, the definition of an episode varies considerably depending on the user, from a brief inpatient stay to a decade of child abuse. It is not appropriate to formalise the definition of an episode at an architectural level, but rather by keeping the fundamental unit of the record as a single Transaction, users will be free to aggregate these as their definitions of episode evolve.

3.5.4 The Transfer of Healthcare Records

The record architecture must allow records to be combined even if they have been held on different computer hardware configurations and on different operating systems.

Transfer of the record, or a part of it, must comprise exchanging whole Transactions. Although a receiving EHCR system will sort these logically into chronological order, it must always be possible to confirm the content of an institution's record as it was at any previous moment.

Transfer of data must be as secure as possible, and references to external data must be maintained on transmission. If parts of the record are "lost" or damaged (for any reason), this fact must be made apparent to future users.

Data must never be assimilated into a clinical record without the involvement of a clinician who is willing to take legal responsibility for that inclusion. "Automatic" amendments by computer systems are not acceptable to clinical users. Test results or other information not yet seen by a responsible clinician should be regarded as external to the EHCR even if held on the same information system.

All Transactions which are part of the EHCR must have an identified responsible healthcare professional. It is this aspect which defines the border of the clinical record.

3.5.5 The Confidentiality of Healthcare Data

Access to the entire healthcare record should be the norm, although there are instances when a consultation record may be abridged at the request of the patient so that some information within it is less openly accessed. The consulting clinician should be able to define the level of access to the whole Transaction, and also have the facility to make individual entries within it more secure. It should be possible to mark potentially upsetting information so as to minimise the risk of its appearing on the screen inappropriately.

Any representations of the clinical data which reflect limited user access, whether for clinical or managerial reasons, must not allow original clinical meanings to be distorted. Transactions must not be fragmented.

An institution receiving a clinical record must be aware of the security access framework of the donor site, and be capable of mapping this into their own such framework. The security status of different staff should ideally be consistent at a national or European level.

3.5.6 Making Use of a Shared Clinical Record

Any entry in the record must be assigned to a Transaction whose author is explicitly defined. Each individual Transaction must incorporate the name of the responsible healthcare professional (who may not always be the author), the date and time of creation or of amendment, the department or institution holding that EHCR, and the Transaction type. The identification of the responsible

HCP must be internationally recognisable, and their professional status must be clear to any future reader.

Users should be able to identify the protocols, templates and decision support tools which are available for their own use with the record. If clinical data has been entered in a previous consultation using terms or classifications not held locally, the user must be made aware of the potential distortion in analysis, translation or scanning which may arise.

3.5.7 Viewing a Clinical Record

The Nota Bene Transaction type is intended to be an overriding alert of key information (warning/priority messages) all users should see before taking clinical decisions. It must be displayed to a clinical user of the record at the outset of that recording or viewing session.

It should be possible to ascertain which triggers are pending for different professionals, and to clearly identify those which apply to the current user and to that institution. Triggers may sometimes only be relevant to one institution: if so they should still be visible as part of the record wherever it is shared, but possibly in an inactive form. This would prevent a patient being recalled for a screening procedure simultaneously by several institutions.

Although individual EHCR systems will display healthcare data differently, it should be possible to obtain an overview of all of the data held within one Transaction, in its original language. This view should reproduce, as accurately as possible, the way that the Transaction was displayed at the time of its creation.

3.5.8 Confirming which language has been used

It must be clear what language has been used in the original recording process. Any Transaction must always be held and available in its original language. A system-translated Transaction must not be allowed to constitute a new version of the original Transaction, and these translations must never be saved as part of the record. The user must always be able to see the original of any Transaction or term.

3.5.9 Confirming which classification system has been used

Where an author has entered information from a term set, that set name (and version) must also be included. The mechanism for coding terms in the record must be implemented in such a way that future versions of such dictionaries carry no risk of incorrectly decoding those records made using earlier versions of a term set. This might be by recording the rubric as well as the code for a term.

Users should be capable of generating intra-record and cross-records analyses, for clinical or management audit purposes. The use of different term sets/languages and translation is an issue when considering queries. No aggregate data covering more than one patient may reside in any

one of their records, unless it has a direct bearing on that patient, and is clearly labelled as being aggregated.

3.5.10 Recording a Transaction

A new Transaction will incorporate the identifier of the responsible clinician, the date/time of the Transaction, and the type of the Transaction. If an amendment has occurred, then the amendment version, the identifier of the amending clinician and the date/time of the amendment will also be documented. The language used for recording the Transaction should also be documented.

The responsible person should be unambiguously identified. If, for example, a secretary was to enter a clinician's notes into the record, the Transactions would be authored by the secretary and authorised for entry by the clinician. A similar authorisation could apply to students' notes.

3.5.11 Recording clinical data

Healthcare information may include a term from a relevant term set, free text, a drawing or diagram, an image or photograph, a biological signal, sound, video or other data type. All such information must only be entered within a defined Transaction.

Each data element must be capable of incorporating certain attributes such as codes from a source classification, locations of other external data element(s), screen display features (including appearance and position such as 'underlined' for medico-legal accountability), protocol or decision support references. Other specific details may emerge in the future, and mechanisms must exist for new attributes to be shareably defined.

Terms should be entered into a Transaction within a context which preserves the meaning, and allows computations. Whether as part of a classification system or not, users must be able to qualify any term with negative, probability or severity statements. Probability might be a percentage or a term, and certainty or severity might be a term or a scale. The qualification NOT should be applicable to any term.

In a system, a set of default units may be assumed by the clinician. In the actual record, the actual units used should always be recorded. The system may guide the user in the recording of local units, but they should not be included in statistics unless conversion formula(e) are provided and approved to map to 'preferred' units. The system should also record the name (or other source) of the scale of the units. If local units are transmitted it must always be clear where the unit is from and that it is not local to the new system.

The transmission of accuracy could be important, and normal ranges may need to be transmitted. Normal ranges cannot always be generated by the system because they depend on context.

Where calculated information is used for clinical decisions, such as a Body Mass Index (based on height and weight), the actual result rather than the formula must be stored in the record. If feasible, the formula should be held in the definition of the term by the system. If a BMI

calculated today is used for a clinical decision, this must not be able to change invisibly through changes made in other Transactions (e.g. the correction of an erroneous weight).

The user must always be free to record a piece of information in free text if structured options appear unsuitable. This free-text might be a single word or a long narrative consultation. It must be possible to associate a free-text comment with a term set entry or with another data type.

The user must be able to define links between any two data elements, without limit to the number of these. Links should be confined within a patient record, and must be managed in a way which supports the transfer of Transactions between EHCR sources.

While recording a Transaction it will often be necessary to scan back through the record, to locate specific Transactions or terms, and to get summaries and graphic displays.

3.5.12 Protocols, templates and decision support

A template is a “static” sequence of some data to be entered. A protocol is a “dynamic” sequence. It is effectively a template which can branch automatically or with user control - it informs as decisions are suggested and taken. The use of protocols must be facilitated by the record.

If diagrams, drawings, tables or graphs are used these structures as well as the data content must be retained in the record. If protocols are used it must be possible to document this within the record. If templates and protocols are to be transmitted to other systems, they must be recognised by the recipient. Data should not be made available to a future user without any accompanying template structures if these give meaning to that data.

Clarity is needed regarding the position of a clinician making a decision based on incorrect decision support information. It must be possible to record which decision support tool was used during a consultation. It should also be clear at a later date which version of the decision support tool was available to the user.

3.5.13 Local term sets and units

The details of any classification system actually used in a Transaction should be recorded.

If local terms are allowed, the set from which the local term came must be recorded. The term set must never be deleted.

When transferring local units or term sets the receiving institution must understand everything that is sent. This could be achieved by transferring the local term/unit by text (in the original language). However, it must be possible for the institution to request the unrecognised term set/unit source from the sender explicitly. There must be a mechanism for uniquely identifying the originator of 'locally' defined terms/ units etc. when these are not local to the current system.

The record and associated dictionaries must be able to accommodate future evolution in

classification systems, and the creation of new terms.

3.5.14 Amending a Transaction

At the time of saving a Transaction, a “point of commitment”, it must become unchangeable, and be indelibly preserved. The Transaction, including all relationships between terms and any attributes of the individual data elements must all be saved unambiguously.

Transactions must be permanent. Once committed by the person making the record, they may be amended but not erased. Amendments must therefore be new Transaction versions. For the purposes of allowing safe and legitimate amendment of details, each Transaction version must document the amending responsible HCP and an amendment date and time. For daily work purposes the older versions may be hidden if the user wishes, making the record complete while only providing relevant information. When transferring the clinical record, only the most recent amendment of any Transaction might be sent. Mistakes, or details which the patient did not agree to have in the record, would therefore not automatically be propagated.

All versions of a Transaction must remain part of the record at the EHCR source which created it. It is only on transmission of the record to others that the user may elect not to forward earlier versions.

3.5.15 Death of a Patient

It must be possible to record details relating to the cause of death, bereavement counselling information given, and possible future legal issues as they arise.

The record should be inactivated at the time of death so that it is not possible to request it in the usual way. However there should be a method of accessing records for medico-legal purposes, research purposes and to counsel relatives.

4 Overview of the GEHR Architecture

4.1 Background

The Deliverables described in Chapter 3 provided the basis of a first attempt to define a formal data architecture, in largely clinical terms: the Interim GEHR Architecture, Deliverable 10 [3]. This had the objectives of providing flexibility to accommodate all potential individual styles of recording and of proposing for evaluation a set of constructs with which to model the data and concepts used in clinical practice.

The remaining work of the project has led to a considerable refinement of the IGA and to the formal expression of the final Architecture contained in this document.

This Chapter introduces the fundamental constructs of this final Architecture and the role which they have in representing the aspects of the EHCR necessary to meet all the requirements described in the previous chapter.

All information in a given HCR implicitly relates to the care of one person, the patient. Within each patient record, the GEHR architecture preserves both the original structure of the data and how the entries in the record are grouped.

Clinicians value the facility for individual expression and creativity within the EHCR. However, this may make it more difficult to share the record. In order to ensure meaning is preserved when the record is transferred from one system to another, GEHR specifies that information should be recorded within its context; the original language and term sets should be identified; and the original structure of the data should be retained (to maintain the correct grouping of specific pieces of information).

The content of a clinical record must be clearly and accountably defined: information should not form part of the record until a HCP has taken responsibility for that information. New information entered into a computer system only becomes part of the electronic healthcare record when it has been committed to the record by an authorised HCP.

Every effort has been made to propose an architecture which is as generic, flexible and non prescriptive as possible. However, where clinicians have identified the need to be prescriptive (for example, in situations where medico-legal security must be maintained) the architecture incorporates features which may be utilised for this purpose.

4.2 Summary of Principal Architectural Components

The principal architectural components evolved by GEHR may be summarised as:

- the EHCR
 - provides the container for all data about a particular patient
- the Transaction
 - provides the majority 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

Each of these is further elaborated using Attributes which address aspects of identification, content and context. They are consistent with the structures apparent in existing records and fulfil the requirements identified by the project for the EHCR.

4.3 Description of the Principal Architectural Components

4.3.1 EHCR

The EHCR is the electronic record for one patient on one system (which is regarded as an EHCR_SOURCE). There is only one EHCR for each patient at this EHCR_SOURCE. Everything which is contained in this EHCR is deemed to be about the patient except when modified by a Collection - see below. This aspect of "being about something or someone", which embodies the idea of 'data subject', is called the 'Scope' of the data here.

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.

4.3.2 Transaction

Definition

A key clinical requirement is an 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 the Functional Specification [7], where 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. Seven different types of Transaction have been identified and have been described more fully in the previous chapter:

- Contact
- Admin.
- Report
- Summary
- Continuing Care
- Nota Bene
- Trigger

Transactions do not contain other Transactions.

Unit of Transfer

In order that the electronic healthcare record 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.

GEHR recognises 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 in the GEHR context which pertain to the same patient.

Such a logical global record is sometimes called the “Virtual Record”. GEHR does not stipulate if or how such a logical record should be realised. It does, however, provide a specification for an Exchange Format (Chapter7) which is considered to be an essential part of any wider system in which EHCRs from different sites are communicated and combined. Chapter 7 also makes recommendations about transferring parts of EHCRs consisting of complete Transactions using the EHCR_EXTRACT concept.



FIGURE 3 The Transaction

Amendment

Transactions are permanent. Once committed by the appropriate HCP, they may be amended but not erased. GEHR proposes a formal amendment concept for Transactions where a “Versioned Transaction” contains all its versions which result from formal amendment. Each version is called a “Transaction Version”.

An amendment, e.g. to correct errors, will result in an additional version of an existing Versioned Transaction, whereas the addition of new information always results in a completely new Versioned Transaction.

Audit Trail

For all types of Transactions, a unique Transaction identification must be generated and stored with the details of who committed the Transaction to the record, and when. GEHR provides this unique identification of Transactions as required for medico-legal purposes, its version control features efficiently accommodating subsequent correction or amendment of Transactions. This same version control allows for the correct, logical merging of EHCR instances referred to above.

In this way a sure and complete chronicle can be made of the evolution of an EHCR.

The scrutiny of Transactions for medico-legal purposes should always take place at their original source where there is a responsibility to retain all historic versions of a Transaction. There is no guarantee that at any other site, to which EHCR data are communicated, there is a complete

account of the data that were present at the original site at the time when clinical decisions were taken there. This could come about because of the possibility that only some Transactions are requested for transfer.

4.3.3 The Organisation of Healthcare Data within Transactions

The Transaction has been described as containing the grouping of information entered about a patient during one recording session. The information within a Transaction would rarely be a single clinical observation, but is more commonly a mixed grouping of clinical findings organized under headings. These headings often convey the situation in which the information was gathered (such as “new patient check” or “antenatal booking examination”) or the relationship it has in time, place or person to the patient (such as “past history”, “management plan”, “laboratory test results” or “family history”). They do not, however, place any constraints or conditions on the clinical data they in turn contain. This concept of structural organization within the record is referred to as *annotation* within GEHR, and the architecture construct to accommodate this information is the *Heading*.

The actual clinical information under each Heading will vary considerably with different situations, and may include historic information from the patient, clinical findings, the results of tests (which may be multi-media), procedures performed, management plans and recall dates, medication intended or actually given, letters or reports, or the results of analyses in the form of tables or graphs. All of this actual clinical data (as opposed to the Headings) is referred to in GEHR as *Observations*.

There will be occasions when the clinical Observations under a Heading will be a single term or value, such as:

<u>Heading: Investigations</u>		
	Peak flow	= 420 l/min

These elemental Observations, comprising a name (such as “peak flow”) and a content value (in this case, with accompanying units) are handled by the architecture construct *Health Record Item (HRI)*.

This patient might have had a couple of extra investigations that day:

<u>Heading: Investigations</u>		
	HRI: Peak flow	= 420 l/min
	HRI: Urinalysis	= normal
	HRI: BM stix	= 2-4 mmol/l

Each of these observations is independent of the others, and can therefore be recorded using HRIs, grouped for convenience under the Heading of “Investigations”.

In many situations more complex clinical concepts will be recorded in which the individual clinical values are interdependent, for example in a hierarchy.

Heading: Physical Examination

Abdomen:

Tenderness:

Location = right upper zone

Guarding = present

Mass:

Location = right lower zone

Size = large

Tenderness = absent

In this example, it is critical that the two specified locations are associated with their appropriate data subjects: tenderness and the mass. It is essential for both tenderness and the mass to be clearly identified as being in the abdomen. It should also be noted that no tenderness was found in the mass itself: in this case the data subject is the mass and not the whole abdomen. This hierarchical construction, which defines the immediate data subject of the information contained within it, is called the *HRI Collection*. Collections may contain additional Collections, HRIs or a mixture of the two. In the above example the outer Collection “Abdomen” would contain two subordinate Collections: “Tenderness” and “Mass”. The remaining details would be recorded as HRIs.

It should be noted that in this example the Heading “physical examination” confirms, for example, that this information did not arise as part of the history from the patient. It does not, however, alter the clinical concepts contained within it: the actual abdominal findings.

It is an important aspect of the use of Headings that they themselves do not constitute a data subject and therefore do not alter the clinical concepts they contain. In the example below:

Heading: Family history**Collection: Father**Heading: Post mortem finding**Collection: Liver**

HRI: Weight= 17 Kg

the Collection structure ensures that 17 Kg is interpreted as the weight of a liver and not of a person, and that the liver is of the patient’s father. The Heading “post mortem finding” conveys the context in which the information was derived, but does not alter the fact that the data subject for the liver is the patient’s father. The term “Family history” provides a convenient Heading, but is not critical since the Collection it contains ensures that the data subject is not the patient but his/her father.

Collections with their subordinate HRIs and/or Collections are used to express the component parts (called Observations in GEHR) of clinical concepts in the correct structural relationship appropriate to the clinical concept, and to assign values to their component parts.

Just as “Observations” is the collective name given to HRIs and HRI Collections, the collective name given to Observations and Headings is “EHCR Entry”. These names have been chosen carefully to keep a close association between the names of Architectural components and words which are commonly used to describe what may be in a record.

Remembering that Transactions occur in EHCRs and that the ‘scope’ of the EHCR is the patient, the starting point for any data within a Transaction is that its scope is the patient; i.e. the data are about the patient. The Transaction itself does not change the scope of the data: only an HRI Collection can do that.

The structure of any Transaction can thus be summarised as: “one or more Observations, optionally annotated by Headings”.

The concepts of the Health Record Item, the HRI Collection and the Heading are described individually in more detail below.

4.3.4 Health Record Item

While data can be entered in HCRs 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. Within GEHR, this has been given a specific name: the Health Record Item or HRI.

Traditionally, individual patient records are built by adding entries at the appropriate location in the relevant record [3]. 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.

The Health Record Item (HRI) provides the mechanism for expressing the content value of Entries made in the record. The HRI does not change the scope of the data.

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

The HRI has been adopted by CEN TC/251 (PT011) as the basic unit of health information within the record. 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. “Weight - 78 kg” and “Diagnosis - Hypertension” are simple examples.



FIGURE 4 The Health Record Item (HRI)

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);
- they represent characteristics of the data subject;
- they derive some of their meaning from the higher level structures to which they belong;

The content of a HRI can be 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 Entries 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).

The values observed are sometimes further defined by some value-specific properties, such as the units in which a value is expressed. This structure can be either implicit, buried in a long sentence of free text (or string of characters):

“a tumour of 2cm in diameter in the right flank”

or explicit, with the various properties of the Observation well segregated and identified:

“tumour:
size: 2 cm
location: right flank”

Most entries in paper HCRs are presented as unstructured data or with a high degree of implicit structure. Computer systems must either record these as unstructured, or capture the structure in a more explicit way.

4.3.5 Health Record Item Collection

The Collection provides a mechanism for changing the Scope of the data. Collections may contain other Collections and HRIs. The lowest level of Collection contains only HRIs.

Collections with their subordinate HRIs and/or 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 general term *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 the decomposition of “blood pressure” into a “systolic” and “diastolic” component, or the detailed description of a set of hearing test measurements. The recursive structure of the Collection allows the HRIs to be assembled into completely flexible but valid structures.

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

The GEHR Collection is similar to the CEN TC/251 (PT011) Health Record Item Complex. However, CEN has not yet distinguished between the two concepts of Collection and Heading, 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 GEHR Collection lead to the unambiguous definition of the data subject of a group of observations, and no explicit data subject attribute is required in GEHR.



FIGURE 5 The HRI Collection

Within Collections, the original organisation of the data can also show some diversity. One can have for instance a simple succession of the HRIs: the ordering of the data can be random or it may reflect the chronological order in which information was obtained and entered. It is important that the structure of the information can be kept and reproduced, even if these same data can also be incorporated in a variety of views [5, 7].

4.3.6 Heading

The Heading provides a means of grouping or labelling 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 labelling or grouping is called Annotation in the GEHR Object Model, clearly to distinguish it from all other combinational devices. Headings do not change the Scope of the data.

Please see Section 4.3.8 for some guidance on how the choice between HRIs, Collections, and Headings is made to represent any clinical concept and its relationship to the patient in an EHCR.

4.3.7 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 health care:
 - 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 an 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, new information is provided by creating new relations between the data. The data can be viewed according to the initial structure, or according to other structures emanating from these links.

4.3.8 The Choice of Constructs to represent Clinical data

This section offers guidance on how the choice of available constructs would be made in order to represent clinical concepts and relate them to the patient in an EHCR

It is not a definition of how an Application is expected to `work' nor how it is supposed to present choices to the user. It is meant to show the parts played by the various constructs appearing in the model from the viewpoint of a theoretical `model-user'.

We start from the mindset of the `model-user' who has some information to enter, having chosen an EHCR and Transaction. The model implies that the following questions are asked in the order specified, and choices made:

Q1 Can the Clinical concept be expressed without changing the Scope of the data?

If not, use a Collection.

Q2 Can the information be expressed as an Observation with a single content value (of a type available in EHCR_INFO)?

If so, use an HRI.

The content value can only be of one type, but the Multi_text Class allows the content to be any combination of free text and coded terms from a termset. Whatever the type of the content value, the content and context comment attributes allow further qualification.

If not, continue to Question 3.

Q3 Can the information be represented by a series of HRIs which are independent of each other?

If so, use a series of HRIs.

Else, use a Collection.

The Collection may have attributes added to convey extra qualifying data which applies to all the members and to the context of this Collection.

The Collection may need to have other Collections nested inside it if the compound nature of the data is sufficiently complex. The Collection which is at the deepest level of nesting contains only HRIs.

Apart from allowing the Scope of the data to be changed, the role of the Collection is to bind together its constituent parts. Each Collection can have its own context attributes which apply to the whole Collection. Each HRI within a Collection may have its own attributes to elaborate further its content and context.

Q4 Does the information in any combination of Collections and HRIs and subordinate Collections and HRIs need to be labelled?

If so, Use a Heading.

Headings annotate: the concept requires that the things to be annotated are “there”. It is convenient to think of the structural assembly of the data being done first, then adding the annotations. (Remember that this is not meant to imply that an application would offer it to the user in this order.)

The Heading can have the role of grouping HRIs provided that the Scope of the data does not need to change. In terms of the analogies used in the past, the Collection-HRI binding is `glue', whereas the Heading annotation is `paint': i.e. glue it together first then paint it!

4.3.9 Effects of Available Termsets on the Choice of Constructs

The naming of Headings, Collections and HRIs is dependent on the availability of suitable terms from termsets.

It is a fact of life that no available termsets are purely elemental, and many contain highly composed terms. This has an influence on the choice of constructs used to represent the data.

The GEHR Architecture allows for the representation of EHCR data using construct names which are of any compositional complexity.

In general if elemental terms are available, then the Collection with subordinate HRIs (and possibly other subordinate Collections) will play a dominant part. If highly compositional naming terms are used, the HRI will be play the dominant role.

The remainder of this section gives some examples of clinical concepts and their representation by HRIs, Collections, and Headings to illustrate the effects of available terms for naming constructs.

In the examples which follow, the following notation is used to represent the use of terms.

<term>	a term_ref where the term may consist of more than one word, but must not be subdivided into individual words.
<term,term,...>	a term_ref with qualifiers
<term>,<term>,...	a multi_text type of content
H:<term>	a Heading
C:<term>	a Collection
HRI:<term> value	an HRI with its content value.

In order to present these examples, it is necessary to allude to particular kinds of attributes such as cx_comment and ct_comment to show how additional information can be given about the context and content of HRIs and Collections. Please see Chapter 6 for a complete definition of all attributes.

Example 1a

Details of some blood pressure measurements using elemental terms.

```
H: <Medical examination>
  C:<blood pressure>
    HRI:<systole>      120 mmHg
    HRI:<diastole>    80 mmHg
    HRI:<site>        <arm, left, upper>
    HRI:<method>      <cuff>
  C:<blood flow>
    HRI:<systole>      20 ml/min
    HRI:<diastole>    13 ml/min
    HRI:<site>        <index finger, left, tip>
```

Example 1b

Same example as 1a, but not explicitly naming the site and method aspects

```
H: <Medical examination>
  C:<blood pressure>
    HRI:<systole>      120 mmHg
    cx_comment <arm, left, upper>,<cuff>
    HRI:<diastole>    80 mmHg
    cx_comment <arm, left, upper>,<cuff>
  C:<blood flow>
    HRI:<systole>      20 ml/min
    cx_comment <index finger, left, tip>
    HRI:<diastole>    13 ml/min
    cx_comment <index finger, left, tip>
```

Example 1c

Same example as 1b, but using context comment for the whole Collection.

```
H: <Medical examination>
  C:<blood pressure>
    HRI:<systole>      120 mmHg
    HRI:<diastole>    80 mmHg
    cx_comment <arm, left, upper>,<cuff>
  C:<blood flow>
    HRI:<systole>      20 ml/min
    HRI:<diastole>    13 ml/min
    cx_comment <index finger, left, tip>
```

Example 2

Same example as 1 but using more composite terms

```
H: <Medical examination>
  HRI:<systolic blood pressure>      120 mmHg
      cx_comment                     <upper left arm>,<cuff>
  HRI:<diastolic blood pressure>     80 mmHg
      cx_comment                     <upper left arm>,<cuff>
  HRI:<systolic blood flow>          20 ml/min
      cx_comment                     <left index finger, tip>
  HRI:<diastolic blood flow>        13 ml/min
      cx_comment                     <left index finger, tip>
```

This shows the dominant role of HRIs when the naming terms are more composite.

Example 3

The weight of the liver of the patient's father.

```
C: <father>
  C: <liver>
    HRI: <weight>      4.6 lb.
```

Note that this can never be interpreted as the weight of the patient, nor as the weight of the patient's liver because each Collection changes the scope of the data.

Example 4

Family History of diabetes mellitus in the patient's father and grandfather.

Three possible solutions are offered:

Example 4a:

Using the Collection to change the scope

```
H: <Family History>
  C: <father>
    HRI: <diagnosis>   <diabetes mellitus>
  C: <grandfather>
    HRI: <diagnosis>   <diabetes mellitus>
```

Note that the change of scope brought about by the Collections ensures that the diagnosis is related to the father and grandfather but not to the patient. It is of course the family history of the patient.

Example 4b: Using only an HRI

HRI: <family history> <diabetes mellitus>
 ct_comment <father>,<grandfather>

Note that it is (also correctly) not possible to infer diabetes of the patient.

Example 4c: If the user/developer insists on starting with diabetes

C: <diabetes mellitus>
 HRI: <family history> <father>
 HRI: <family history> <grandfather>

It is important to realise the subtlety of the meaning of this construction:

1. It cannot be inferred that the patient has diabetes because <diabetes mellitus> in this case is only a Collection name, and it is not possible to decide whether a) this was a diagnosis of diabetes b) the patient thought he had diabetes or c) the doctor thought the patient might have diabetes, or anything else.
2. Rather, the Collection introduces a new scope or data subject into the record: The subject is diabetes mellitus, and what follows in the HRIs is information about diabetes mellitus.

Although this somewhat contorted example is safe from wrongful medico-legal interpretation, it is far preferable to use 4a or 4b.

5 GEHR Object Model Classes

This Chapter provides an informal description of the classes of the GEHR Object Model. It also contains some explanation of how it is expected that the model will be used in practice.

For a formal description, please see the following Chapter which contains a list of all the classes, derived programmatically from the Eiffel source texts of the classes, with a diagram for each cluster. The class texts also contain additional comments to aid understanding of the attributes of each class.

The organisation of classes into clusters reflects the conventions of the Eiffel compiler which was used to verify the syntactical correctness of the classes and their inter-relationships. They also provide a convenient grouping of the classes.

5.1 EHCR Cluster

This cluster models various concepts relating to EHCRs themselves.

(This section contains information about how an application implementing this architecture is expected to manage the exchange of EHCR data. Some access and security issues are also addressed. The information is given for guidance only, and is included in order to explain the features of the “top-level” classes included in the model.)

The EHCR itself represents the health care record for a patient, in electronic form, and is the central concept of the GEHR information model. The `dt_creation` attribute of the EHCR class identifies the point in time when the EHCR began its life (a medico-legal requirement).

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.

The LIST of Transactions must contain VERSIONED_TRANSs. Each VERSIONED_TRANS must contain at least one TRANS_VERSION (the latest one).

5.1.1 Acquisition Requests

Given the basic restriction of exchanging only logically “complete” Transactions, EHCR information is acquired by the destination requesting an EHCR_EXTRACT. When sending or acquiring an EHCR_EXTRACT the following properties should be specified:

- the desired subset of versioned Transactions
- which versions of each Transaction to send/keep:
 - all versions
 - just the most recent version

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. As there is no global patient identifier that can be modelled, the conflict may be resolved by comparing the latest version of the **subject** attribute in the most recent ADMIN Transaction (see Version Control, Section 5.3.1 and ADMIN, Section 5.3.4). If there is any doubt, the final decision must be left to the person responsible for accepting the record at the receiving site.

5.1.2 Access and Security

It is important to realise that for the most part, security is an issue for systems and applications: it is about controlling the usage of information, rather than an intrinsic property of information. The information model is only capable of providing features for enabling security measures to be implemented by applications and systems.

From the point of view of the model, only a single basic access control mechanism is provided: access/amendment permissions. The main situation envisaged is that information known to be “sensitive” at the time of recording needs to be marked in some way so as to enable access control.

Access restriction should be used with care. It may be dangerous to limit access to parts of a Transaction, if this were to alter the understanding gained by a viewer who then proceeds to make decisions which may be different from those based on knowledge of the full Transaction. It is therefore important that systems always indicate clearly if the user has a restricted view.

The classes for whom permissions apply are currently thought to be:

- HCP authorising
- HCP legally responsible for the care of the patient
- other HCP
- other staff member
- patient
- other persons

and any combination of the above is allowed.

In some cases there is a desire to allow certain classes of HCP access to certain kinds of information in the EHCR, e.g. making blood pressures only available to a particular class of HCP. It is currently felt that this requirement is one of intelligent filtering of information rather than blocking or granting access, and is best achieved by the use of queries in applications.

5.2 Identification of Persons in the EHCR

The need to record the identity of various persons concerned with “making” the EHCR has been identified in the Requirements. The information model provides attributes in appropriate classes to contain the necessary identification details. This section provides a list of such persons and some relevant background information.

- HCP Authorizing the Entry into the record
This is the most important of these attributes, and is to be regarded as mandatory. It will be recorded at the Transaction Version level, and is of type HCP.
see the hcp_authorizing attribute.
- HCP legally responsible for the care of the patient at the time that this Entry was made
This is provided to enable applications to track the legal responsibility for care of the patient.
It will be recorded at the Transaction Version level, and is of type HCP.
see the hcp_legally_resp attribute.
- Information provider
This attribute will be used if the source of the data entered into the record is not the HCP_authorizing. Typically this might be another HCP in the same or another Health Care Facility, the patient or a family member.
It will be recorded at the Observation (i.e.HRI or HRI_Collection) level, and is of type PERSON.
see the hcp_info_prov attribute.
- HCP contacted
This attribute will be used to record the HCP with whom a face-to-face contact took place.
see the contact_with attribute in Contact Transaction
- Recorder
This will enable the actual staff member who entered the data into the record to be identified. It may or may not be the same as the HCP_authorizing, but, if not, is considered to be acting with the approval of the HCP_authorizing e.g. a medical secretary.
It will be recorded at the EHCR_Entry (i.e. HRI or HRI_Collection or Heading) level, and is of type Staff_member.
see the recorder attribute.

- Shadow Author

This is provided so satisfy the educational requirement for the submission of data for inclusion in the EHCR by students (Medical, Nursing etc.)

It is envisaged that the committal of such data will be made, as usual, by a responsible HCP. Applications could use the information that the data came from a student for a number of purposes, for example, (i) the omission of such data from some types of statistical returns; (ii) finding and reviewing student data with the student.

It will be recorded at the EHCR_Entry (i.e. HRI or HRI_Collection or Heading) level, and is of type Person.

Note:

Data recorded by the patient (e.g. at home on to a smart card) or by a member of his family for the purpose of providing logging information will only become part of the EHCR when committed by a responsible HCP. The shadow_auth attribute will not be used to identify the source of the data: rather the info_prov attribute will be used (see Observation Class).

5.3 Transaction Cluster

The GEHR concept of “Transaction” is described in Section 4.3.2 and Deliverable 7 [7] and should not be confused with the database management system notion of physical Transactions. A Transaction corresponds to an interaction with the EHCR by one HCP and 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.

Transactions and changes to Transactions actually correspond to two different types of change to the EHCR:

- creation of a new logical Transaction due to a contact, report etc. where new information is added;
- creation of a new version of the information in an existing Transaction - an amendment.

5.3.1 Version Control

A simple linear versioning scheme¹ is proposed having the following characteristics:

- a new version can only be made from the most recent version;
- the contents of previous versions are never altered after committal;
- each version is uniquely identified with a systematic revision identifier;
- it is possible to access any version by specifying a revision identifier;

- it is possible to save some information about each version when it is created;
- it is possible to generate a difference report between two versions of a Transaction by specifying the relevant revision identifiers;
- versions can never be deleted;
- it is possible to obtain a summary of the versions of a particular Transaction;
- it is possible to determine what versions of the Transactions constituted the EHCR for a particular date/time in the past (medico-legal requirement).

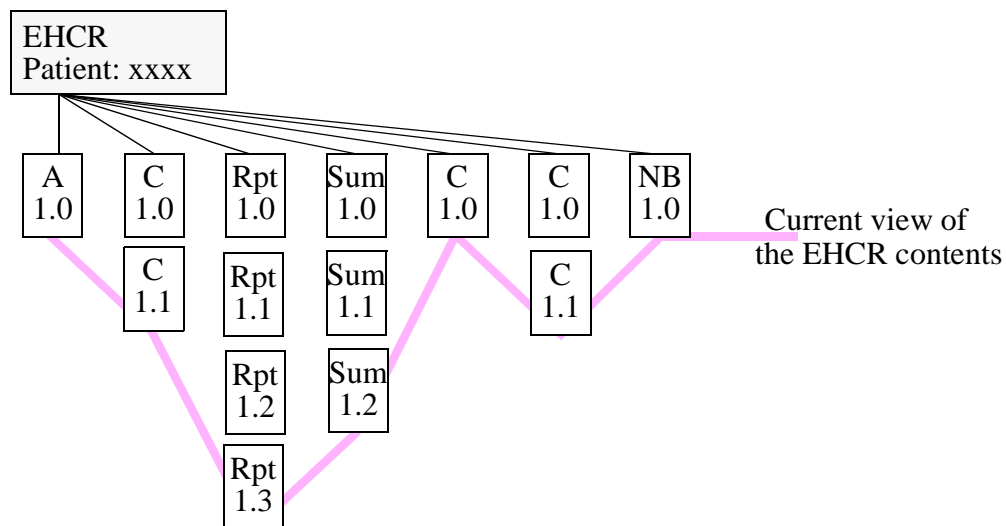


FIGURE 6 EHCR with Transactions and Version Control

These requirements form a subset of various well-known version control paradigms, including SCCS (UNIX Source Code Control System), RCS (Revision Control System; UNIX, PC), CMS (Code Management System; VAX), and numerous object version control systems in ODBMS products (e.g. ObjectStore, Versant etc.). This paradigm may be applied to Transactions to give the idea of a versioned Transaction, as illustrated by Figure 7.

The notion of “committing” is associated with each version, rather than with the Transaction as a whole, since the Transaction version is the unit of modification.

The idea of “merging” is associated with entire versioned Transactions, not with single versions.

1. In this section the language used to describe how version control might work appears to specify the operational characteristics of some application system. The reader is asked to remember that this architectural description (model) is not of the application system, but of the structure and content of the data within the EHCR. The actual details of how such a version control scheme would be implemented are not appropriate here: the model does, however, have classes and attributes which describe the minimum necessary functionality of the proposed version control scheme.

The version control mechanism does not change the motivation for creating new Transactions: a new Transaction is created for a new contact, report, summary etc. A new version in an existing Transaction is only created to correct errors or omissions, not to add new information.

Version control relates to the Transaction information as follows: Imagine that some information is created for addition to the record, and consider the information as the first version of a potentially many-versioned Transaction. The act of committing the version will do two things:

1. create a `VERSIONED_TRANSACTION`. Information which is constant for all versions of the Transaction is included here as attributes, such as amending rights and date/time of creation;
2. create a `TRANS_VERSION` containing the information, as well as some logging information indicating the revision identifier, date/time of committal, and the HCP involved.

Note that the model of version control used here says nothing about implementation, in particular mechanisms for compression of versions by storing “changes” or “deltas”. Efficient data compression does not enter into the semantics of the model of version control in any way: logically every version is the complete contents of the Transaction at a point in time.

Note also that (because of the linear version control) it is not permitted to amend a Transaction other than at its source. In the simple case where a clinician at another site felt the need to amend an acquired Transaction, the request should be given to the source. Although this situation is currently believed to be rare, there are more complex situations of this nature which will be a subject of future work.

One, more likely, situation is where a HCP wishes to record, say, an opinion about a former Transaction within the EHCR to which the original Transaction is transferred. It is recommended that the HCP should create a Report Transaction to contain the opinion, expressed in the usual way using Headings and Observations. This Report Transaction can then use Link Attributes on its Observations to make the association with the relevant Observation(s) in the original Transaction.

5.3.2 VERSIONED_TRANS

The transactions to be found in an EHCR at a particular Source site will either have been created there or will have been transferred from another site. (see also Chapter 7)

In the case where a Transaction is at its original site, it is obligatory to record the date and time at which the Transaction was committed to the record as well as the identity of the recorder and of the HCP who authorized the entry of the data into the record.

In the other case, where the Transaction came from elsewhere, an alternative form of the Transaction - the `ACQUIRED_VERSIONED_TRANS` - is used to hold additional attributes for identifying where the Transaction came from, and what its identity was in the other place. The Acquired version of the Transaction also records the identity of the HCP responsible for acquiring it.

The combination of the `source` + `source_trans_ref` attributes uniquely identifies the Transaction's first version at its original site.

If a Transaction has been acquired from another `EHCR_SOURCE`, the `source`, `hcp_auth_acq`, `source_trans_ref` and `was_gehr_source` attributes will be non-void.

Amending rights are defined at the `VERSIONED_TRANS` level and (like all attributes of `VERSIONED_TRANS`) may not be amended. Access rights (the rights to view the data) are also defined in `VERSIONED_TRANS`, but subsequent amendments to the Transaction may put stricter access rights on individual Entries within the Transaction.

The same `GEHR_Version` is assumed for all versions created under a particular `VERSIONED_TRANS`. This does not imply that previously committed data should be retrospectively changed to a new `GEHR_Version`. This area is partly discussed in Deliverable 15 [13] and will be a subject of Future Work.

5.3.3 TRANS_VERSION

`Revision_ids` should be automatically generated when a new version is created, and are guaranteed to be unique with respect to any other `TRANS_VERSION` in the same `VERSIONED_TRANS`.

The HCP authorising committal of the Transaction is not necessarily the same person as the recorder. The act of authorisation legally implies that this HCP knows and understands the full contents of the Transaction.

The recorder is the person who physically interacted with the EHCR in order to record this Transaction. It must have been on the authority of the `hcp_authorising`, but is not necessarily the same person.

The `STANDARD_TRANS` class models standard Transaction types. So far the only nonstandard Transaction type identified by GEHR is 'Trigger'. The difference between the Transaction types is outlined below. For more detail, refer to Deliverable 7 [7].

5.3.4 ADMIN

Models an Administration Transaction e.g. name, address, insurance details. Every EHCR instance must be unambiguously associated with just one patient in the European context. This is (as far as possible) established by inspecting the most recent version of the `ADMIN` Transaction which is required by the model to exist in every EHCR. This contains the attribute 'subject' which contains the latest name, date-of-birth and gender of the patient (as well as possibly other administrative information). Where possible, the `ehcr_id` of the EHCR can be used to identify the patient's EHCR uniquely within the `EHCR_SOURCE` (see also Acquisition Requests, Section 5.1.1 and Version Control, Section 5.3.1).

5.3.5 CONT_CARE

The contents of a Continuing Care Transaction describe future care planned. This Transaction type may be seen as complementary to the SUMMARY Transaction type.

The responsible clinician would normally select those Entries which he/she feels are appropriate to describe the plan of care to other clinicians. The period covered by the plan may be recorded as an attribute.

5.3.6 CONTACT

A Contact Transaction is created for a contact (or encounter) with the patient. The date and time of the occurrence of the contact should be recorded. The 'contact' may have been a consultation with the patient, a home visit or perhaps a phone-call but always relates to a single date and time at which this 'contact' took place. If such a contact did not occur, a Report Transaction would be appropriate.

5.3.7 NOTA_BENE

The Nota Bene Transaction is used by a clinician to record those Entries which he/she believes are important to be seen on first accessing the patient's record. If it is felt that this 'list' should be changed, a new NOTA_BENE should be created with the clinician's new list of important Entries. A system might offer the previous Entries as an initial list from which to select.

5.3.8 REPORT

The contents of a Report Transaction can be characterised by having medico-legal status outside the EHCR. Report Transactions may be incoming (such as test-results) or outgoing, or may be a report made by an HCP about the patient but without a contact with the patient. If the report is a reply to an earlier request, this association may be recorded.

5.3.9 SUMMARY

A Summary Transaction relates to past care up to the moment of committal. This Transaction type is complementary to the CONT_CARE Transaction. A clinician might select some of the Entries from those recorded in other Transactions.

5.3.10 TRIGGER

The Trigger Transaction is the place in which actions (e.g. recalls, reminders) may be recorded as a result of various conditions being true (e.g. high blood pressure, missed appointment). As yet, no detailed model has been developed. This will be a subject of future work and this class acts purely as a place-holder for later results.

5.4 Item Cluster

The Item Cluster includes classes which provide the structure of health care information in standard Transactions, reflecting its use by practitioners.

The intention is to model two general kinds of information:

- “observations” (includes measurements, facts, hypotheses, etc.): This information is seen as having a well-defined structure, and as being complete in the medical sense - no further information is needed to convey the “facts” as observed, measured or described by the practitioner or patient. This information is also independent of the pattern or form of the clinical session or event causing its creation. (see also section 4.3.3)
- “subjective” headings: included as annotations to the objective information. They provide additional contextual information about the relationship between the data and the patient in time, place, or person. They have no structure of their own, except where nesting occurs. They are used by the practitioner to organise the observation information, and their use reflects the structure or style of the clinical event, and the care model of the practitioner.

For example, the observations, however complex, recorded under the heading “Past History” are no more or less complete in themselves if the heading is omitted: the heading serves to add further context, and may well be useful in carrying out retrospective analyses of the data. The GEHR architecture prescribes that the names of HRIs, HRI Collections, and Headings are terms from a term set. GEHR supports the use of local term sets for this purpose.

The “observation” information is modelled by the Observation, HRI_COLLECTION and HRI classes, while the Heading class models headings. Figure 8 illustrates a general example of Observation and Heading information, with respect to Transactions.

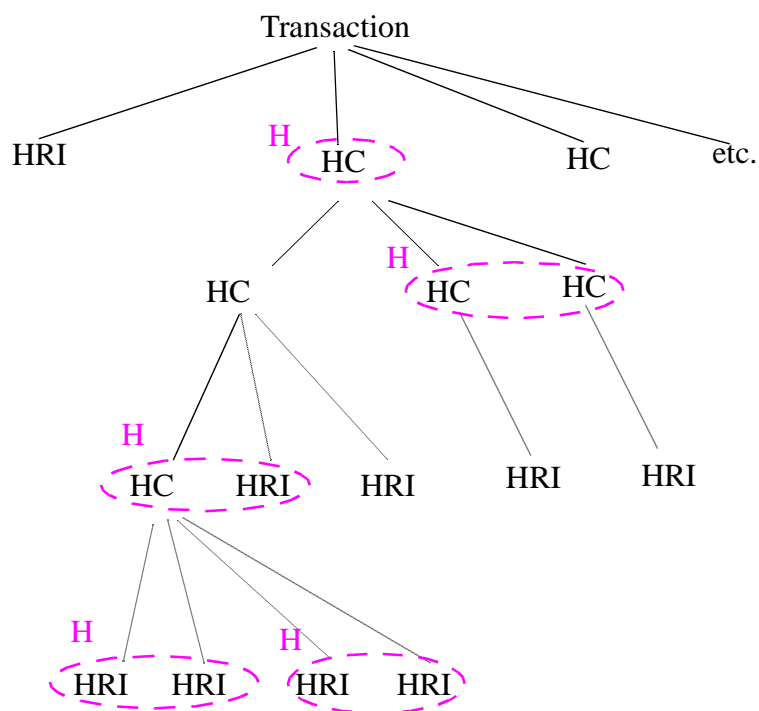


FIGURE 7 EHCR_ENTRY General Structure

From the formal information point of view, several things are worth noting in this model:

- the model describes the structure of underlying information, which is consistent and complete, not the structure of a particular view;
- all classes receive some basic attributes from EHCR_ENTRY via inheritance. In particular, the **recorder** attribute appears here, enabling HEADINGS, HRI_COLLECTIONS and HRIs to be individually attributed to a recorder;
- Headings can only annotate “sibling” Observations i.e. HRI_COLLECTIONS and HRIs at the same level in the same hierarchy (see Figure 9) (this is the only approach that makes sense);
- nested Headings are allowed, i.e. more than one Heading can annotate the same Observation;
- to achieve arbitrary Headings, local term sets can be used.
- one way of understanding the Observation inheritance sub-hierarchy is to realise that HRI_COLLECTIONS act as 'qualifiers' for HRI information. For instance an HRI representing the quantity “size = 3 cm” may be qualified by HRI_COLLECTIONS

representing respectively “tumour”, “thigh (left, upper)”. Without these, the HRI on its own is ambiguous, and therefore useless;

Although the “correctness” of the Observation information has been stressed here, the model is designed only to enable the creation of “correct” information. The actual medical information models required to do this, e.g. anatomical models, disease information models, hearing test models, viral information models are completely outside the scope of the GOM. These medical information models are the domain of applications and ultimately of practitioners themselves. Thus the GOM makes no claim to be able to check or enforce the correctness of this information.

Figure 9 illustrates the combined use of Heading and Observation.

Note that a Heading could have a value (name) which appears to be the same as the name of a HRI_COLLECTION or an HRI, since the semantics are those of description and qualification respectively.

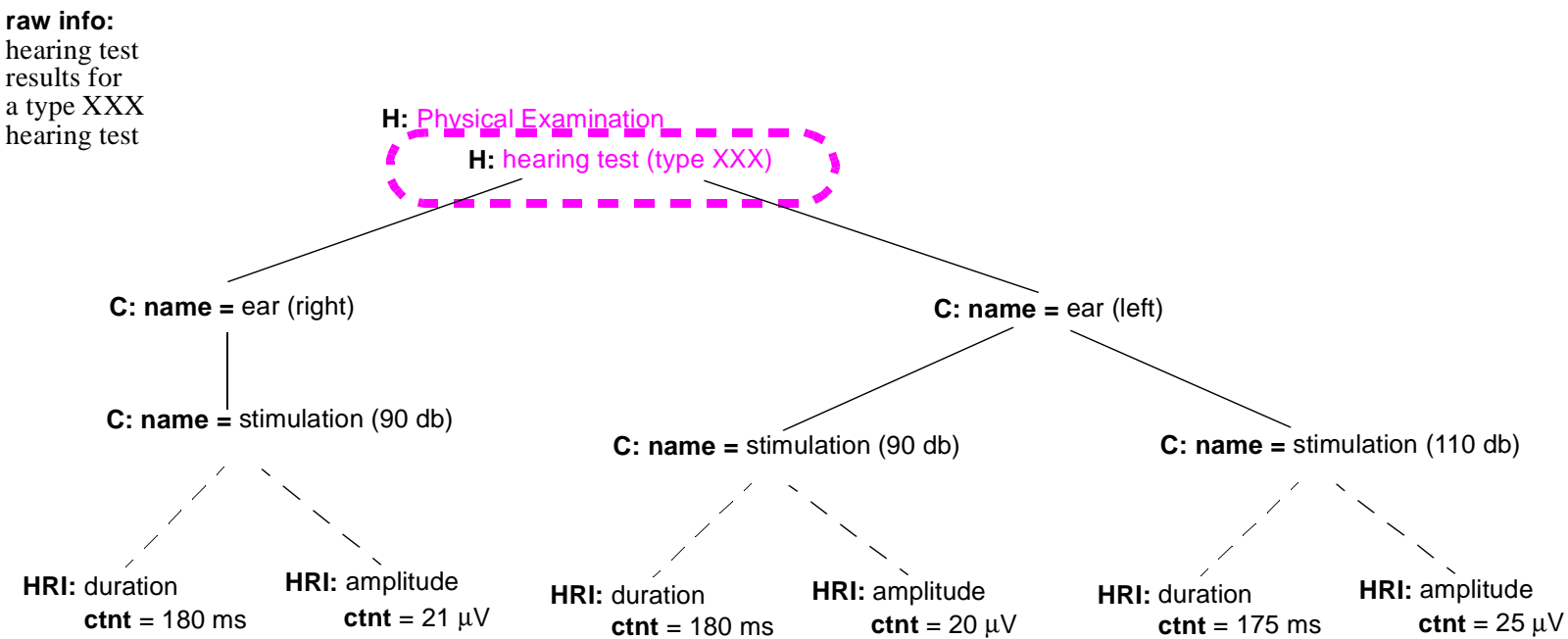


FIGURE 8 HRI and Collection Example with Headings

Some fairly simple rules and implications of building **HEADING** and **HRI/HRI_COLLECTION** information structures can be stated:

- the amount of qualification needed is the minimum which will unambiguously identify the information with respect to the patient, since care is provided with respect to patients (i.e. not to one shoulder, or to a cancer condition). Thus, while a practitioner might simply record “blood pressure”, the implied meaning is “blood pressure of the patient”. This “qualification” is defined by a medical information model, as mentioned above;
- the idea of information about other persons, such as the patient's mother can be easily represented as follows:
 - create a separate **HRI_COLLECTION**;
 - set the **name** to indicate the person, e.g. “mother”;
 - if appropriate, link it to the **HRI_COLLECTION** or **HRI** it is related to (e.g. Next-Of-Kin) using the **link** attribute.

This keeps **HRI_COLLECTION**s relating to other people separate from those relating to the patient, minimising risk of confusion.

There may be many ways of creating an Observation structure to represent the same concept. As long as the final structure is completely unambiguous with respect to the patient when read by an HCP, the particular structure chosen is immaterial. The choice of possible structures will probably be guided by available terms.

The person recording the Entry (**recorder**) may wish to emphasise it. In an EHCR implementation this might be achieved by underlining or perhaps by use of a different colour. Several basic levels of emphasis are defined by GEHR and may be recorded in the **emphasis** attribute of **EHCR_ENTRY**. If it is desired that individual parts of an Entry are emphasised differently, the **ct_emphasis** attribute of **HRI** may be used.

In the situation where a student has added his/her own notes, and these have become part of the official record by virtue of an authorising HCP, the **shadow_auth** attribute is used to identify the student. (The **recorder** may still be different e.g. a secretary) The issue of student (‘shadow’) notes is discussed in Deliverable 9 [11].

5.4.1 HEADING

This class has the purpose of annotating one or more **HRIs** and/or **Collections**. A **Heading** assists a clinician to arrange data in a way which suits his/her personal preference.

5.4.2 HRI

Models a “leaf” Observation, i.e. one which cannot contain other Observations.

The content of an **HRI** is an instance of **EHCR_INFO** (and therefore any subclass). Note that **TERM_REFS** with multiple qualifiers are already catered for by the **qualifiers** attribute of

TERM_REF.Comments are handled by the `ct_comment` attribute of this class. It is currently thought that outside these cases, there is no need for multiple Entries.

Occasionally a clinician may wish to record a specific comment about the context of the recording (e.g. “over the telephone”, “instrument was faulty”). The `cx_comment` can be used for this.

The `dt_observed` attribute may be used to record the date/time that the Observation took place in cases where this may be distinct from the date/time of occurrence (for contacts) or of committal. For example, the date and time at which a test took place as opposed to the date and time the result is entered in the record.

The `certainty` attribute acts as a place-holder for the recording of a level of certainty about an Observation. This may currently be any term or free text, but note that unless the `certainty` is void (by default assumed to be ‘certain’), or the term is a globally recognised one, it may be difficult guarantee that the correct interpretation is conveyed (see discussion on the Evolution of Term Sets, Section 8.3). The complex area on the recording of uncertainty will be a subject of future work.

The `is_derived` attributes indicates whether the information was derived by some computation process rather than directly entered by a person, for example BMI. This may be used to provide legal protection for an HCP under some circumstances.

5.4.3 OBSERVATION

`info_prov` identifies the person actually providing the information. Normally this is the same as the `hcp_authorising` of the owning Transaction, but sometimes, e.g. in cases where the information is received from outside, not as part of an EHCR, but as perhaps test results, a report etc., the provider's identity is recorded here. This attribute can also be used to record the identity of other persons providing data for the record e.g. the patient himself, or a member of the patient's family.

5.4.4 HRI_COLLECTION

This class is the containment structure for HRIs. It behaves as qualification for the contained HRIs. At any given level the `HRI_COLLECTION` may comprise HRIs and/or `HRI_COLLECTIONS`.

The Collection also alters the scope (i.e. data subject) of the contained data.

See Chapter 4 for more information.

5.5 EHCR_info Cluster

5.5.1 EHCR_INFO

Models an item of EHCR information. Items of known types are modelled by appropriate subclasses,

Emphasis might apply separately to each part of ECHR information. Emphasis levels currently defined are low, medium and high, of which the default is low. Implementations are free to indicate medium or high emphasis by graphical or other means (e.g. bold, colour)

The purpose of this cluster is to provide a model of “real” information which will formally define certain well understood types of information, such as terms and quantities, while ensuring that arbitrary information types, such as multimedia, can be accommodated in an open sense.

From this point down, the GOM models only generic information concepts, rather than information usage concepts. Thus classes contain only features which would be generically applicable in any context - even one unrelated to the EHCR.

5.5.2 Generic Data Types

The structure of this part of the model should reflect the balance between accepted and contentious concepts about generic data types at any point in time. Accepted concepts may be explicitly modelled using classes which define minimum semantics, while concepts for which there is no consensus are modelled in a generic way. The difference is that those things explicitly modelled can be transparently understood and manipulated by any conformant implementation, no matter what site the information came from, whereas very few things can be assumed about generically modelled things, and they may be handled in any way that works. As time goes on, and the medical and information communities converge on more accepted definitions for certain concepts, the model can be improved by providing explicit definitions, while retaining generic ones for some time.

The BOOL class is used when only a true/false (or yes/no etc.) value is needed.

5.6 Text Cluster

A MULTI_TEXT instance contains a LIST of PLAIN_TEXTs, which means that the actual items in the LIST could be PLAIN_TEXT or TERM_REF.

The PLAIN_TEXT class models any single piece of text e.g. text which may be as typed in freely by a user. A special form is when the text is a term from a termset. It should not be confused with WP text containing formatting information, which is a multimedia type. A large piece of (even unformatted) text should probably be referenced as a bulky data (multimedia) Item.

The orig_lang attribute specifies the language in which this piece of information was originally

recorded. This may be important to differentiate between nuances of meaning when information is transmitted across language borders. Note that this means that contents, names of EHCR_ENTRYs, and also HRI comments, will all have original language recorded for them.

Since the TERM_REF types inherit from PLAIN_TEXT, they can be used wherever a PLAIN_TEXT is specified.

A TERM_REF has additional attributes to complete its role as a carrier of structured textual information:

- a code for the concept's preferred term, if one exists
- a code representing the term used e.g. if a synonym was used instead of the preferred term corresponding to this code from this term set, or if a locally-defined term set was used
- identification of the term set used
- optional further qualifier terms
- the plurality of the term.

The attribute `concept_code` is used to represent a recognised clinical concept which the user wishes to record e.g. the code for "myocardial infarction". The user may choose to record the preferred term, but, alternatively, may record a synonym such as "heart attack". The `code_used` attribute is used for this purpose and for recording codes from locally defined term sets.

If the preferred term is used, it may not be necessary to use the `code_used` attribute, which would then be left void. In the case where a locally registered term set is used, the terms may not necessarily represent standardised clinical concepts i.e. they may not correspond across term sets: `concept_code` would be void but `code_used` would have a non-void value in this case. Locally registered codes could be used to preserve exactly the original intention in cases of shades of meaning causing doubt amongst terms and supposedly compatible synonyms.

Many medical terms are likely to be qualified in practice, but this may depend on the nature of the term set used.

The plurality of the noun implied by the code value is given in the attribute `is_plural`. It is used to generate the correct spellings of plural versions of nouns. Most languages are unsystematic for this, for instance:

- (English): bone/bones but foot/feet
- (French): patient/patients but hospital/hopitaux

5.6.1 TERMSET_DESC

This class models a term set descriptor. A code set has a name, and may be local or it may be registered with an agency. If the name is of a local term set (which is therefore not registered), the name will be the same as the actual term set name; if it is a registered term set, the name will be the registered name.

For example this attribute might have the value “57” if this were the CEN-registered name for, say, “ICD_10”.

Codes for registered code sets can be used anywhere the code set is itself available. This means GEHR information need only include the codes, and meanings will always be generated locally (in the local language) via the local copy of the code set.

For locally-defined term sets the name of the registering agency would be “LOCAL”. A recognised string is used for registering agencies e.g. “GEHR”, “CEN”, “LOCAL”.

5.6.2 TERM_SET

This class models the minimum requirements of a Term Set in the GEHR context. This includes the name, revision_id and term_sets. The terms_used (modelled as a TABLE) will require the ability to look up a textual representation of a code in the appropriate language.

The minimum model is defined where the table has two dimensions, both compound. The first dimension is an aggregate of the concept code and code used. The second dimension corresponds to an inner table of the corresponding expanded terms in one or more languages.

The precise details of how a table lookup is to be performed are left to the results of other projects and future work. (See also the Evolution of Term Sets, Section 8.3)

5.7 Quantity Cluster

5.7.1 QUANTITY

Models a quantity with value and optional units. The quantity may or may not be measured. Quantities such as “120 mmHg”, “2.5 m”, “0.14 m s⁻²”, “2 h 43 min 18 s” and “55 per minute” can be modelled.

The precision attribute can be used to specify the number of significant figures (if is_sig_figs is TRUE) or decimal places (if is_sig_figs is FALSE) to which the value has been recorded. If this is not known, these attributes should remain void.

If units are recorded, they are in the form of a list of pairs (unit_id, exponent). For example: 10 g dL⁻¹ would be recorded as value 10, with units (<g>, 1) (<dL>, -1).

The `unit_id` corresponds to a UNIT about which at least the term is known, and normally more information (see UNIT cluster)

For vector style units e.g. 10 st 2 lb 3 oz, the quantity should be recorded as the value in the lowest unit (e.g. 2275 oz) with the attribute `is_style_single` set to **FALSE**.

If the quantity is measured by an instrument, it is possible to record this in the `instrument` attribute. Values such as “by eye”, “manually” etc. are also acceptable.

The attributes `accuracy` and `expr_as_pc` can be used to indicate whether the value of `accuracy` is a percentage. e.g. 2275 oz \pm 3 oz would have `accuracy` = 3, `expr_as_pc` = **FALSE**, 2275 oz \pm 0.5% would have `accuracy` 0.5, `expr_as_pc` = **TRUE**.

5.7.2 QTY_RATIO

This class models a ratio of two quantities such as “250 mg/500 ml”. Since QUANTITY has attributes `num_prop` and `den_prop`, it is also possible to model “250 mg solute per 500 ml solvent”. This appears to be the only real use for QTY_RATIO so far. Note that a QTY_RATIO instance expressing the solute/solvent idea implies different semantics from an equivalent QUANTITY expressing solution strength: the QTY_RATIO is expressing materials for making something, and implies a certain final volume, whereas the other is simply the concentration of the final (or any other) solution.

5.7.3 Q_RANGE

A range of quantities (measured or not) is modelled by the Q_RANGE hierarchy. Any range with two limits, and any combination of inclusion of the limit values, as well as the sense of the range (within or without) can be modelled. Single-sided ranges are modelled by setting one limit to the infinity value (positive or negative). Note that `max` is never less than `min`.

5.8 Units Cluster

Recognised Units are modelled by the UNIT class. Each Unit is either a BASE_UNIT (from a BASE_UNIT_GROUP) or a DERIVED_UNIT (from a DERIVED_UNIT_GROUP). Base Unit Groups are one of:

mass, length, time, thermodynamic temperature, luminous intensity, electrical current, amount of substance, plane angle or solid angle (the latter two are technically supplementary unit groups but are handled by the same class).

Examples of each of BASE_UNITS are therefore:

mass:	kg, stones, oz
length:	m, km, inches
time:	days, s
thermodynamic temperature:	K, °C

luminous intensity:	candela
electrical current:	amp
amount of substance:	mole
plane angle:	°, rad
solid angle:	sterad

Each of these units is associated with a Unit System (e.g. S.I., British Imperial)

A DERIVED_UNIT_GROUP is formed from the 'product' of BASE_UNIT_GROUPS. For example:

Volume is a derived unit group whose **derivation** is (<length> , 3)

Pressure is a derived unit group whose **derivation** is (<mass>, 1) (<length>, -1) (<time>, -2)

Examples of DERIVED_UNITS are therefore:

Volume:	Litres, cc
Pressure:	Pa, psi

If a derived unit has no single term (e.g. s^{-2}), its 'derivation' (e.g. (<s>,-2)) is stored in the 'units' part of the QUANTITY sub-class (see QUANTITY above).

In order to convert quantities with the `is_style_single` attribute set to false, it is necessary to know a number of things about the unit of recording.

Firstly, where it appears in a hierarchy of related units (e.g. where lb appears in relation to st or oz, or where minute appears in relation to hours and seconds). This is recorded in the `next_larger` and `next_smaller` attributes. Where the unit is at one end of the hierarchy, the appropriate attribute may be left void. e.g. for oz, `next_larger` would be pounds, but `next_smaller` would be void.

Secondly, some units are not normally used when expressing a vector quantity e.g. 1115 ml would normally be shown as 1 L 115 mL as opposed to 1 L 1 dL 15 mL or 1 L 1 dL 1 cL 5 mL. Units (such as dL and cL in this example) that are not standard for expressing vector quantities have the attribute `is_standard` set to false.

Finally, to perform the conversion between e.g. oz and lb or seconds and minutes or indeed between units from different unit systems (e.g. miles to km), an implementation will need a table with columns of the form `from_unit`, `to_unit`, `conversion_formula`. Conversion formulae are generally linear and so a multiplication factor and additional constant will usually suffice. Non-linear scales will require more complex formulae.

5.9 Bulky Data Cluster

It is always assumed that bulky data are opaque, whether a reference to a physical object, a piece of multimedia or an unrecognised ('alien') object. Electronic data requires a reference (see URI below) and a `logical_type` (e.g. 'Drawing', 'sound'). The only additional thing required to be

known for a piece of multimedia is the data format (e.g. JPEG, GIF etc.). Using this information, a receiver of such information can interpret the data assuming it has tools capable of processing the specified format.

Much the same is true of so-called alien objects, except that the format of the data is also unknown. References to such data may be held in the record but it is likely that only the source of the information would be able to interpret it. The model includes a **method** attribute in the **ALIEN_DATA** class which contains a pointer to where the interpretive software is held. Such data could, with time, become a recognised piece of multimedia or indeed a class of its own. It is important that such references are kept in the record even if the data themselves cannot be processed by a particular system.

References to data which are not (or more usually, cannot) be electronic (e.g. a paper file or letter, a specimen) are catered for with the **PHYSICAL_DATA** class.

The **storage_loc** should unambiguously identify the location with respect to the context where the reference exists. If storage is in the same HCF, then the value may be relative to that. If the storage is in another HCF (i.e. the data containing the reference has been transmitted) the location should be globally unique. Examples include “east wing lab 5 cold room”, “Royal Marsden Hospital, Sutton, Physics Dept., Cold room 2”.

The **reference** should be with respect to the **storage_loc**. Example: for shelf storage (of say x-ray slides) there may be a particular numbering system in use for shelving at the location specified by **storage_loc** requiring a **reference** such as “shelf 5 box 17 sample 21”.

The **storage_type** is a guide to the type of storage being used. It is intended for information purposes, and is not needed to resolve the address. Possibilities include “shelf”, “safe”, “cold_store”

5.10 People and Places Cluster

The **PERSON** class has only the absolute minimum detail needed about all persons in the GEHR context. This is currently just the **name**. **NON_PATIENTS** usefully require means of contact. **STAFF_MEMBERS** (who might be recorders in a record) will normally have a **grade** and **position**. The **HCP** class must include mandatory attributes for country of registration and for registration number, and an optional attribute to record profession.

The **PATIENT** class contains the minimum information required to identify the Patient's record when received from another source. Until more global patient identifiers are in use, this is thought to be the **name**, **date-of-birth** and **gender** of the patient. Note that, because this information is recorded in the **ADMIN** Transaction, even if the patient changes his/her name (or perhaps gender), the first **ADMIN** Transaction will still be present in which the information was first recorded. This is also the case if the date-of-birth (for example) was entered wrongly and was then amended - it is still possible to access the original version.

The **HCF** class models the minimum information about a Health Care Facility in the GEHR context. It must have at least a **name**.

The ADDRESS may include any number of lines of text, each denoted by a TERM_REF. Typically these might be addressline1, addressline2..., town/city, county/state, country. The postcode is at present an unchecked string. The address is considered to be valid from the date given.

5.11 Moment Cluster

The DATE_RANGE Class models a range of dates, times or date/times. The date at either end of the range may be included or not and the idea of ‘within’ or ‘outside’ the range can also be recorded. The infinity value (positive, or negative as appropriate) may be used to model a one-sided range.

5.12 GEHR Basic Cluster

Basic classes explicitly model concepts for which there is good agreement for a minimum definition. In an implementation, it is only necessary that the classes be consistent with the model, not the same. In fact implementation classes will often have many more features and relationships.

5.13 Exchange Cluster

The classes described here are seen as primitive classes by GEHR: these are the classes which are the building blocks for all other classes. Therefore they require primitive encoding and decoding exchange rules from which all other decoding methods can be built. These classes are also assumed by the GEHR object model to be provided by implementation libraries. Thus the only semantics required of these classes by GEHR are:

- they exist;
- the implementation provides minimum semantics implied by the class names. For example, an instance of INTEGER corresponds to an integer number;
- a rule for encoding to and decoding from the exchange format exists;

The semantics described below are not mandatory, but act as a guide for “sensible” implementations. See [19] for an example of a comprehensive semantic definition of these classes.

5.13.1 Universal Resource Identifier (URI)

The URI concept has been proposed (see [32.]) and is in use on the World-Wide Web (WWW) as a means of identifying remote information objects. It assumes nothing about an object except that there exists a means of accessing it. URIs can accommodate the use of different access protocols, and different object types, including directory objects, interactive “objects” (e.g. login sessions)

and objects that can be queried; this is done using the concept of a scheme, which is part of the URI. The basic format of a URI is simple, and is simply a string:

scheme:path[?search]#fragmentid

Typical schemes include “ftp”, “gopher”, “telnet”, “mailto”, “http” and “database”. New schemes may be defined and used, making the URI an open concept with respect to object types. The semantics of the part coming after the “:” are defined completely by the scheme coming before it. Thus the “path” part could be interpreted as a directory path in a file system for scheme “ftp”, or an addressable sub-object in a scheme corresponding to an object data server. An object can be identified using a search string, which must be understood by the scheme, and a fragment of an object can be identified using the fragmented part.

Examples of URIs are:

ftp://info.cern.ch/pub/www/doc/http-spec.txt

{ftp URI of [32] in plain text format}

http://info.cern.ch/hypertext/WWW/Addressing/URL/Overview.html

{http URI of [32] in hypertext format}

gopher://athena.sdsu.edu:71/11/sounds

{gopher URI of a directory of sound files}

gehr_ehcr://barts.ac.uk:82?pat_id="S1930A"+exp_bulky=0+exp_term=1

{query specifying BARTs EHCR for patient
S1930A, with bulky expansion off and term
expansion on}

Experimental schemes must be prefixed with “x-”, and may be used by mutual agreement between parties. It has been proposed that the Internet Assigned Numbers Authority (IANA) perform the function of registration of new schemes.

The use of URI is recommended for use within GEHR, at least for exchanged data (if not local data) since they provide an open-ended identification system for remote objects of any kind, and also because there are already official definitions and tools in existence.

5.13.2 GEHR_UID

GEHR Unique Identifier. Used for remote referencing only.

5.13.3 REVISION_ID

This gives the revision of a version of a version-controlled entity. This class is where version-naming/numbering rules will be formalised as a result of future work. For now, it is proposed that the first version be '1.0' and that subsequent versions increment the first decimal place.

6 Formal Expression of the GEHR Architecture

This Chapter describes the GEHR Object Model (GOM). This is **Version 1.0**

6.1 Introduction

The requirements identified in previous deliverables and outlined in Chapter 3 have informed the development of this information model. Before presenting the object model, the way in which the features within the model relate to the requirements will be discussed.

In the following descriptions, classes are grouped into “clusters” as a means of sub-dividing the model. Clusters are a convenient grouping mechanism used in the Eiffel compiler, but do not have any structural significance of their own in the model.

The formal class texts presented in this Chapter are generated from Eiffel classes using the Eiffel compiler “clickable” option. This gives a shortened view of each class, i.e. only feature interface syntax (name, type, arguments) and semantics (pre- and post-conditions, class invariant). The Eiffel clickable form was then processed into “mark-up language” texts for inclusion in this document.

Figure 10 shows the main classes of the model and their relationships. It is included here to act as a ‘road map’ to the rest of this chapter. Please note that it only shows the “upper” levels of the model down to the level of content type of HRIs. Please see the later parts of Chapter 6 for the remaining classes.

Although the diagrams are based on the “Rumbaugh” methodology, it is not pure Rumbaugh [16] and the following section detailing the extensions should be read carefully.

For this first full version of the GEHR Architecture it was decided to omit from the GOM all features which could be said to be “Implementation-specific”. Thus the classes defined only have attributes as features: they do not contain computed features (functions) as would be expected in a model of this kind. In order to compensate for the lack of functions some dummy attributes were added to the deferred classes and defined lower down the inheritance tree. These additional attributes are present in the source texts which were validated by the Eiffel compiler, but have been omitted for clarity in the class texts and diagrams presented in this Chapter.

This Chapter provides a diagrammatic representation of the GOM interleaved with Eiffel class texts.

The GEHR Exchange Format which was derived from the GOM is presented in the next Chapter.

6.2 Object Modelling Formalism Used

The formalism chosen for this Chapter consists of:

- diagrams based on the “Rumbaugh” methodology ([16]), with concepts added from the Eiffel language ([18]) and the BON notation ([15]);
- an underlying textual representation expressed in the Eiffel language;

Although Rumbaugh diagrams are not truly formal, and do not deal with all of the possible semantics of an OM, such as correctness, they do not prevent the addition of extra semantics to classes and operations. Rumbaugh has been chosen as a basis in the interim, as it is widely known and used, and is not flavoured too heavily by implementation considerations.

Please see Appendix A.3.1 for further comments on the choice of formalism.

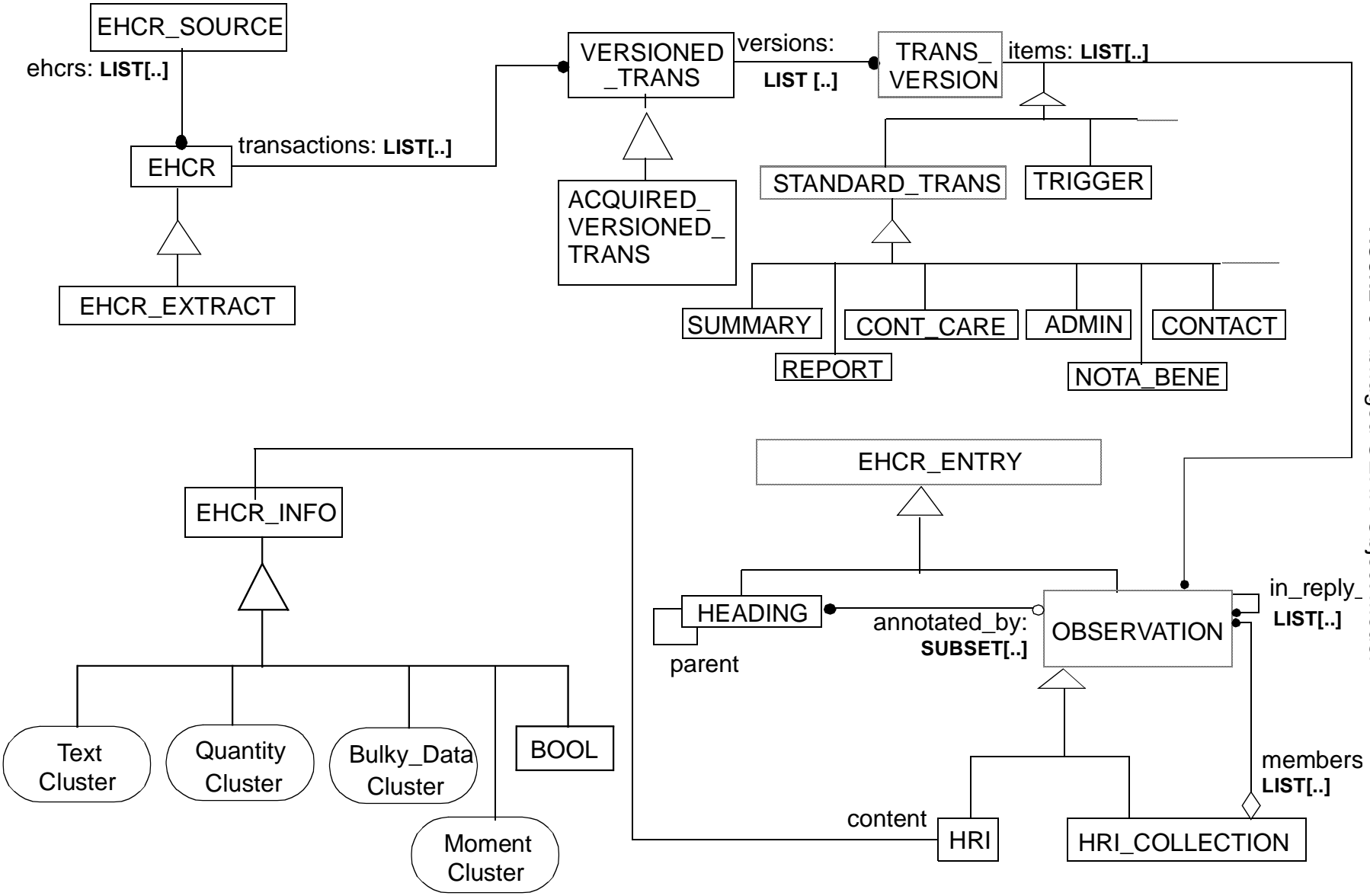


FIGURE 9 Abridged GEHR Object Model

6.2.1 Diagram Conventions

Conventions adopted for the use of Rumbaugh diagrams are as follows:

- class names are in bold upper-case; attribute names are in lower-case;
- where an attribute appears on an association line between two classes, the name is placed closest to the class to which it belongs. This is the *reverse* of some conventions, but is done so that:
 - all attributes of a given class can be easily found clustered around the box for that class;
 - diagrams which are split over multiple pages, such as the model presented here, do not have “floating” attribute names from classes in diagrams on other pages;
- instances can be associated with each other in only two ways:
 - logical reference
 - “in-line” expansion

The vast majority of instance associations are of the first kind, the second usually being reserved for optimizing access of objects such as integers, characters, booleans and so on. For example, in the Eiffel system, only the classes INTEGER, FLOAT, DOUBLE, BOOLEAN, and CHARACTER are normally “expanded” like this (although in fact any class can be expanded if necessary).

References appear in the GOM in two equivalent forms:

1. lines between classes. The reference exists as an attribute in either or both classes; the association line is then annotated with the attribute name(s);
2. the notation **attribute:TYPE** inside a class box has the same semantics. This is often used when an excess of “uninteresting” connections would only clutter the diagrams.

In-line expansion uses the same notation as 2.) above, since semantically there is very little difference.

Note that the logical notion of “reference” is often confused with physical implementations such as “pointer”, “address”. See Appendix A.4.

- Rumbaugh diagrams use hollow or filled balls at the ends of association lines to indicate multiplicity. While these are preserved in this model for the visual effect, the multiplicity is correctly given by the corresponding attribute names, which will usually include a generic container, such as **members:LIST[..]**. This convention is taken from the BON notation, as it is much more precise and also corresponds exactly to the equivalent Eiffel text.
- correctness conditions such as invariants are sometimes abridged on the diagrams so as to reduce clutter; the full version always appears in the relevant class text.

There follows a contents list for the individual clusters and classes described in this Chapter.

EHCR Cluster	90
6.4 EHCR_Source	91
6.5 Class ehcr	91
6.6 Class ehcr_extract	92
Transaction Cluster	94
6.7 Class versioned_trans	95
6.8 Class Acquired_ver_trans	96
6.9 Class trans_version	97
6.10 Class admin	98
6.11 Class cont_care	98
6.12 Class contact	99
6.13 Class nota_bene	99
6.14 Class report	100
6.15 Class standard_trans	100
6.16 Class summary	101
6.17 Class trigger	101
Item Cluster	102
6.18 Class ehcr_entry	103
6.19 Class Observation	104
6.20 Class hri	105
6.21 Class hri_collection	106
6.22 Class heading	106
6.23 Class ehcr_info	109
6.24 Class bool	109
Text Cluster	110
6.25 Class plain_text	111
6.26 Class multi_text	111
6.27 Class term_ref	112
6.28 Class termref_qualifier	113
6.29 Class term_set	114
6.30 Class reg_agency	115
6.31 Class termset_desc	115
Quantity Cluster	117
6.32 Class measurement	118
6.33 Class mqty	118
6.34 Class mqty_range	119
6.35 Class mqty_with_units	119
6.36 Class q_range	119
6.37 Class q_with_units	120
6.38 Class qty_range	121

6.39	Class qty_ratio	122
6.40	Class qty_with_units	122
6.41	Class quantity.....	123
Units Cluster.....		124
6.42	Class base_unit	125
6.43	Class base_unit_group.....	125
6.44	Class derived_unit	125
6.45	Class derived_unit_group	126
6.46	Class unit	126
6.47	Class unit_group	127
6.48	Class unit_system	128
Bulky Data Cluster.....		129
6.49	Class bulky_data	130
6.50	Class electronic_data	130
6.51	Class alien_data	131
6.52	Class multimedia_data.....	131
6.53	Class physical_data	133
Moment Cluster.....		134
6.54	Class date_range	135
6.55	Class occasion	135
Persons and Places Cluster.....		137
6.56	Class person.....	138
6.57	Class non_patient.....	138
6.58	Class patient.....	139
6.59	Class staff_member	139
6.60	Class person_name	140
6.61	Class address	140
6.62	Class hcp	141
6.63	Class hcf.....	142
Basic Cluster.....		143
6.64	Class date_time	144
6.65	Class date	144
6.66	Class time.....	144
6.67	Class moment.....	145
6.68	Class phys_ref	145
Exchange Cluster.....		147
6.69	Class gehr_lang	148
6.70	Class gender_code	148
6.71	Class std_numeric	149

6.72	Class <code>emph_level</code>	149
6.73	Class <code>gehr_uid</code>	149
6.74	Class <code>ehcr_uid</code>	150
6.75	Class <code>trans_uid</code>	150
6.76	Class <code>entry_uid</code>	151
6.77	Class <code>permissions</code>	151
6.78	Class <code>revision</code>	152
6.79	Class <code>code_link</code>	153
6.80	Class <code>uri</code>	153

6.3 GEHR Object Model Diagrams and Class Texts

EHCR Cluster

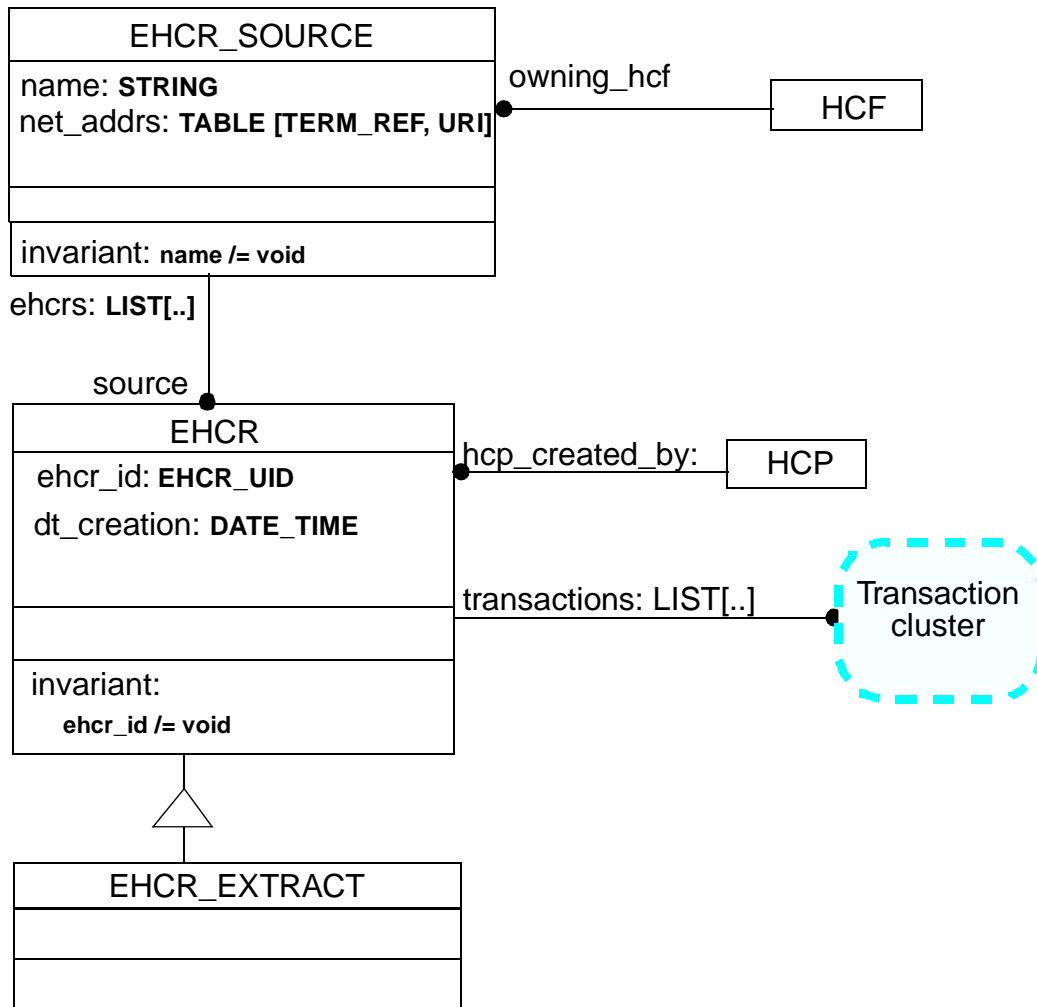


FIGURE 10 EHCR Cluster

6.4 EHCR_Source

```
-- ehcr\EHCR_SOU.E
-- Version 1.0 1995-6-30
--
-- Models a GEHR-compliant EHCR source.
--
```

class

EHCR_SOURCE

feature -- Identification

name: STRING;

-- Name of the EHCR provider. Must be unique within the
-- enclosing context (could be "country", "Europe" etc.).

owning_hcf: HCF;

-- Defines which Healthcare Facility (HCF) owns this EHCR source.
-- All EHCR_SOURCES are part of an HCF. Provides a unique
-- identity of the HCF in which the EHCR was originally
-- created.

net_addrs: TABLE [TERM_REF, URI];

-- Network address(es) of the EHCR_SOURCE
-- The term_ref can be used to identify what sort of network address this is

feature --Content

ehcrs: LIST [EHCR];

-- The EHCRs at this EHCR_SOURCE.

invariant

name /= void;

end -- class *EHCR_SOURCE*

6.5 Class ehcr

```
-- ehcr\EHCR.E
-- Version 1.0 1995-6-30
--
-- Electronic Health Care Record Class. Models an Electronic Health
-- Care 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.
--
```

```
class
```

```
    EHCR
```

```
feature -- Identification
```

```
    ehcr_id: EHCR_UID;
```

```
feature -- Content
```

```
    transactions: LIST [VERSIONED_TRANS];
```

```
feature -- Context
```

```
    dt_creation: DATE_TIME;
```

```
        -- Date/time of creation of this EHCR. Identifies
        -- the point in time when the EHCR began its life
```

```
    hcp_created_by: HCP;
```

```
        -- The HCP responsible for creating the EHCR.
```

```
invariant
```

```
    ehcr_id /= void;
```

```
end -- class EHCR
```

6.6 Class *ehcr_extract*

```
-- ehcr\EHCR_EXT.E
```

```
-- Version 1.0 1995-6-30
```

```
--
```

```
-- To model what is necessary to extract from an EHCR in order to send
-- the information to another EHCR_SOURCE.
```

```
-- An EHCR_EXTRACT looks like an EHCR, with the exception that the
-- LIST of transactions may contain a list VERSIONED_TRANS
-- which is fewer in number than the whole EHCR from which they
-- were extracted.
```

```
-- Nothing is said here about the physical transfer mechanism, but it is
-- known that the version of the GEHR architecture must be transmitted
-- by some means before the receiving EHCR_SOURCE can interpret the
-- contents of the EHCR_EXTRACT received.
```

```
class
```

EHCR_EXTRACT

inherit

EHCR

end -- class *EHCR_EXTRACT*

Transaction Cluster

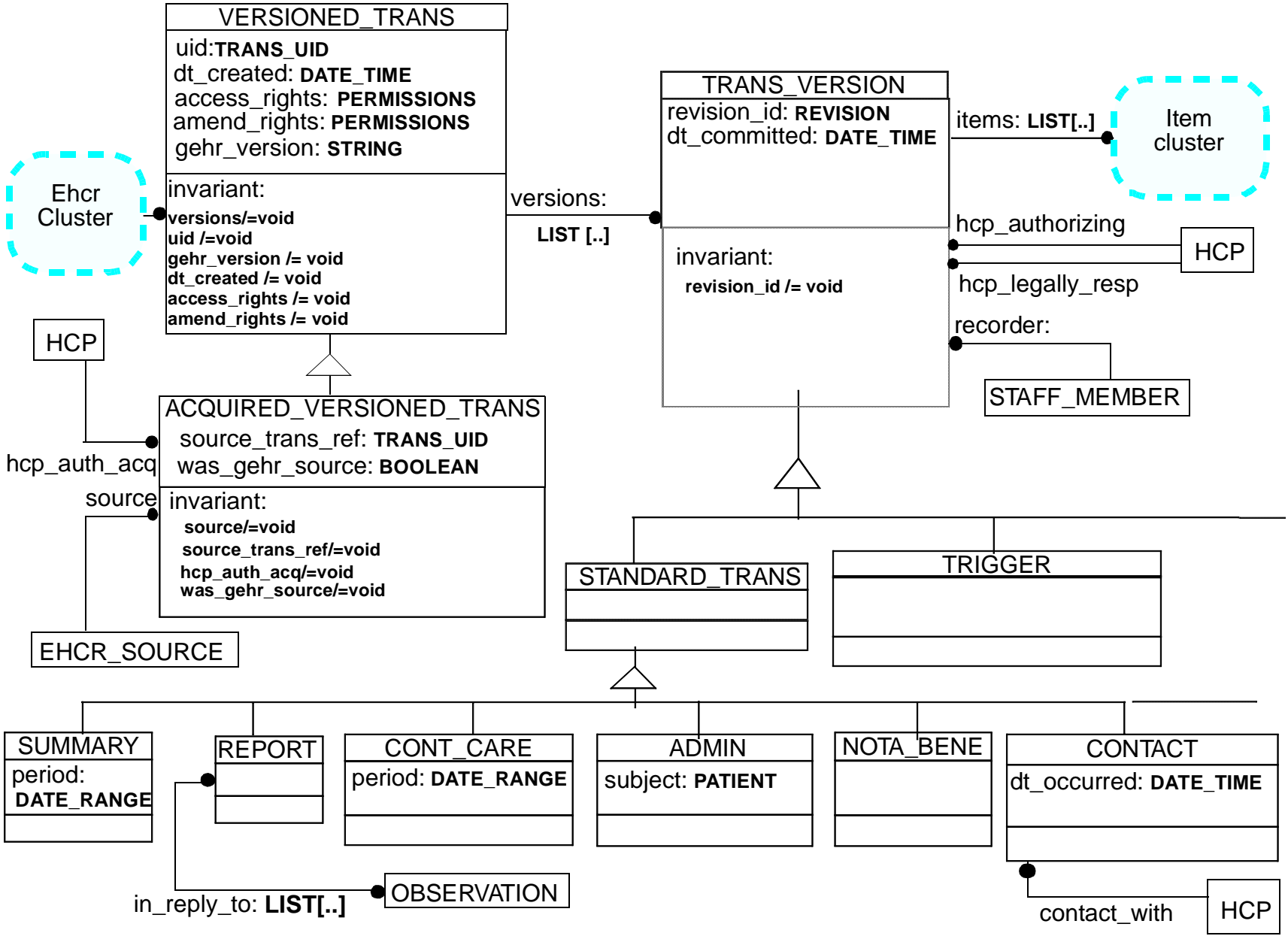


FIGURE 11 Transaction Cluster

6.7 Class `versioned_trans`

```
-- transact\VERSIONE.E
-- Version 1.0 1995-6-30
--
-- Models a logical transaction whose information exists in versions
-- corresponding to the different times at which each piece of
-- information was created. The VERSIONED_TRANS is therefore a
-- mechanism for grouping and controlling the TRANS_VERSIONS.
--
```

class

`VERSIONED_TRANS`

feature -- Identification

`uid: TRANS_UID;`

feature -- Content

`versions: LIST [TRANS_VERSION];`

-- The `TRANS_VERSIONS` associated with this logical
 -- transaction. Provides for a linear model of version
 -- control.

feature -- Context

`dt_created: DATE_TIME;`

-- Date/time at which the transaction was created in this
 -- EHCR (regardless of whether the first version was
 -- acquired or not). Not to be confused with the
 -- date/time committed of a `TRANS_VERSION`, which
 -- corresponds to the committal of actual information.

`access_rights: PERMISSIONS;`

-- Enables (read) access to the information in the
 -- logical transaction to be controlled.

`amend_rights: PERMISSIONS;`

-- Enables the right to amend the information in the logical
 -- transaction to be controlled.

`gehr_version: STRING;`

-- Particularly relevant for exchange and implementations
 -- The version of the GEHR standard at the time of
 -- creation of this `VERSIONED_TRANS`. The same GEHR version
 -- is assumed for all versions created on a given
 -- `VERSIONED_TRANS`.

invariant

versions /= void **and** *uid* /= void **and** *gehr_version* /= void **and** *dt_created* /= void **and** *access_rights* /= void **and** *amend_rights* /= void;

end -- class *VERSIONED_TRANS*

6.8 Class *Acquired_ver_trans*

-- transact\ACQUIRED.E

-- Version 1.0 1995-6-30

--

-- Models that part of a logical transaction whose information is
 -- necessary when the transaction has come from another *EHCR_SOURCE*
 -- as the result of a transfer of data.

--

class

ACQUIRED_VERSIONED_TRANS

inherit

VERSIONED_TRANS

feature

hcp_auth_acq: *HCP*;

-- The HCP authorising acquisition & inclusion of the
 -- information contained in this transaction.

source: *EHCR_SOURCE*;

-- The source of the acquired transaction.

source_trans_ref: *TRANS_UID*;

-- The uid of the transaction at its source.

was_gehr_source: *BOOLEAN*;

-- to indicate whether the *EHCR_SOURCE* of this transaction
 -- was *gehr*-compliant or not.

invariant

hcp_auth_acq /= void **and** *source* /= void **and** *source_trans_ref* /= void **and**
was_gehr_source /= void;

end -- class *ACQUIRED_VERSIONED_TRANS*

6.9 Class trans_version

```
-- transact\TRANS_VE.E
-- Version 1.0 1995-6-30
--
-- Models a particular version of a VERSIONED_TRANS. TRANS_VERSIONs
-- are where the "real" EHCR information is kept.
--
```

class

TRANS_VERSION

feature -- Identification

revision_id: REVISION;

-- Identifies this version. Unique with respect to any other
-- TRANS_VERSION in the same VERSIONED_TRANS.

feature -- Content

items: LIST [OBSERVATION];

-- The information contained in the transaction

feature

dt_committed: DATE_TIME;

-- Non_Void unless the Transaction was received from elsewhere
-- Date/time at which committal of this version actually
-- occurs (regardless of how long the transaction has taken
-- to construct).

hcp_authorizing: HCP;

-- Non_Void unless the Transaction was received from elsewhere
-- The HCP authorizing committal of the transaction. Not
-- necessarily the same person as the recorder. The act of
-- authorization legally implies that this HCP knows and
-- understands the full contents of the transaction.

hcp_legally_resp: HCP;

-- The identity of the HCP who was legally responsible for
-- the care of the patient at the time the transaction
-- information was recorded.

recorder: STAFF_MEMBER;

-- Non_Void unless the Transaction was received from elsewhere
-- Actual person recording the item electronically, by
-- interaction with an application or similar means. To
-- make it clear who physically interacted with the EHCR

```
-- in order to record this item. It must have been on the
-- authority of the hcp_authorising, but is not
-- necessarily the same person (it could be a secretary
-- for example)
```

invariant

```
revision_id /= void
```

```
end -- class TRANS_VERSION
```

6.10 Class admin

```
-- transact\ADMIN.E
-- Version 1.0 1995-6-30
--
-- Models Administration transaction
--
```

class

```
ADMIN
```

inherit

```
STANDARD_TRANS
```

feature --

```
subject: PATIENT;
```

```
-- The patient who is the subject of this EHCR. Every EHCR
-- instance must be unambiguously associated with just one
-- patient in the European context.
```

```
end -- class ADMIN
```

6.11 Class cont_care

```
-- transact\CONT_CAR.E
-- Version 1.0 1995-6-30
--
-- Continuing Care transaction type: the content describes future care
-- planned, and this transaction type may be seen as complementary to
-- the Summary Transaction type.
--
```

class

CONT_CARE

inherit

STANDARD_TRANS

feature -- Access

period: DATE_RANGE;

end -- class *CONT_CARE*

6.12 Class contact

-- transact\CONTACT.E
-- Version 1.0 1995-6-30
--
-- Models Contact transaction type.
--

class

CONTACT

inherit

STANDARD_TRANS

feature -- Context

dt_occurred: DATE_TIME;

contact_with: HCP;

end -- class *CONTACT*

6.13 Class nota_bene

-- transact\NOTA_BEN.E
-- Version 1.3 1995-6-13
--
-- Model Nota-bene transaction type.
--

class

NOTA_BENE

```
inherit
    STANDARD_TRANS
```

```
end -- class NOTA_BENE
```

6.14 Class report

```
-- transact\REPORT_T.E
-- Version 1.0 1995-6-30
--
-- Report transaction.
--
-- A Report transaction may be formed by importing the contents of a
-- report received into the transaction
-- Report transactions may also be used for recording data of the kind
-- which would normally be recorded under a Contact Transaction, but
-- where no contact between the patient and the HCP actually took place.
```

```
class
    REPORT
```

```
inherit
    STANDARD_TRANS
```

```
feature --
```

```
    in_reply_to: LIST[OBSERVATION];
        -- Cross-reference to the observation requesting the report.
```

```
end -- class REPORT
```

6.15 Class standard_trans

```
-- transact\STANDARD.E
-- Version 1.0 1995-6-30
--
-- Model standard transaction types.
```

```
deferred class
    STANDARD_TRANS
```

```
inherit
    TRANS_VERSION
```

```
end -- class STANDARD_TRANS
```

6.16 Class summary

```
-- transact\SUMMARY_E
-- Version 1.0 1995-6-30
--
-- Summary transaction type: the content relates to past care up to this
-- point in time; this transaction type is complementary to the
-- Continuing Care type.
```

```
class
    SUMMARY
```

```
inherit
    STANDARD_TRANS
```

```
feature -- Access
```

```
    period: DATE_RANGE;
        -- Period this summary covers.
```

```
end -- class SUMMARY
```

6.17 Class trigger

```
-- transact\TRIGGER.E
-- Version 1.0 1995-6-30
--
-- Model trigger transactions.
--
```

```
class
    TRIGGER
```

```
inherit
    TRANS_VERSION
```

```
end -- class TRIGGER
```

Item Cluster

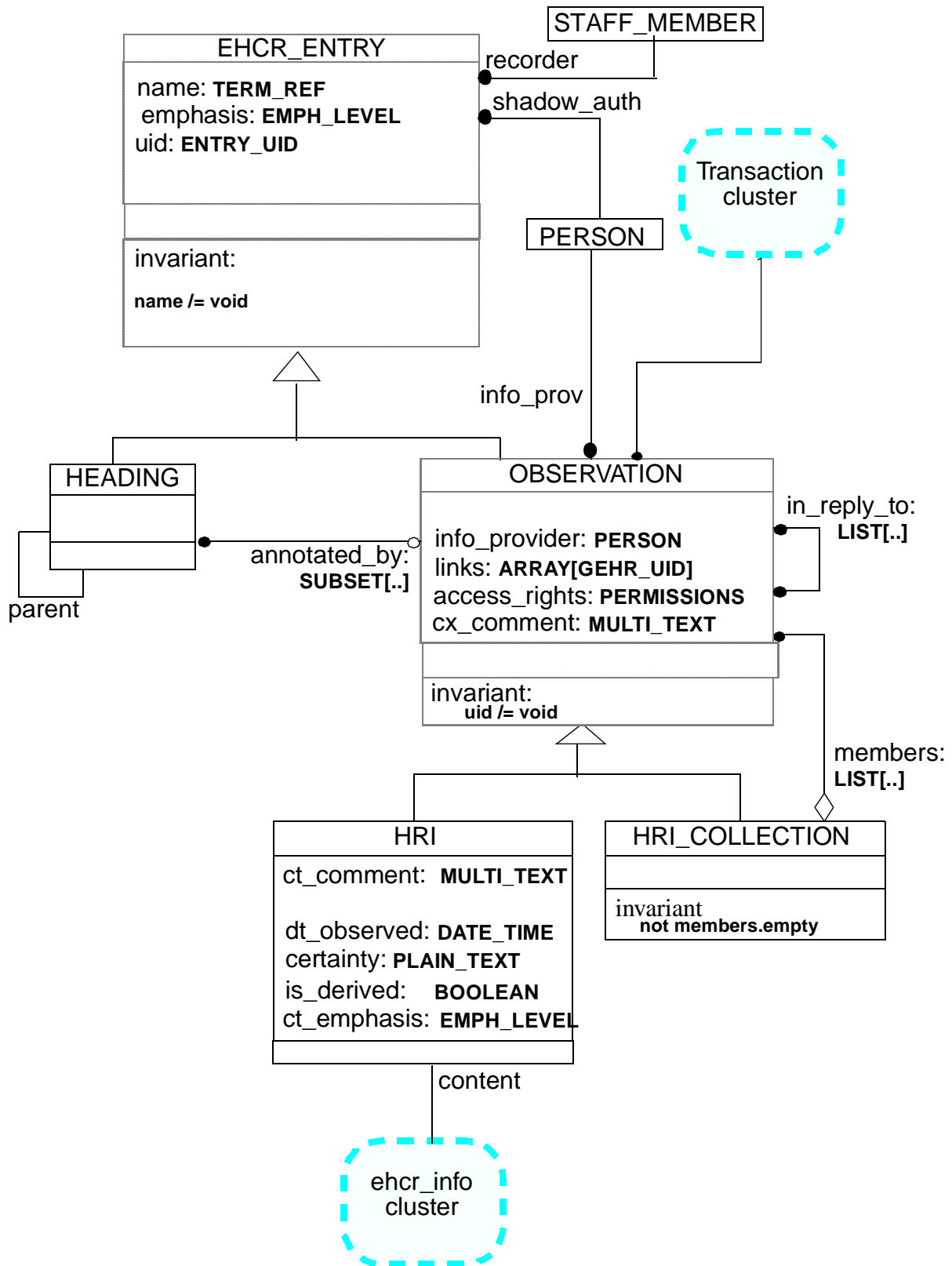


FIGURE 12 EHCR Item Cluster

6.18 Class ehcr_entry

```

-- item\EHCR_ENT.E
-- Version 1.0 1995-6-30
--
-- The EHCR_ENTRY class contains those features expected to occur in
-- all types of entries in EHCRs.
--
--
class
    EHCR_ENTRY

feature -- Identification

    uid: ENTRY_UID;

    name: TERM_REF;
        -- Name of the entry. For HEADING, this is the heading.

feature

    emphasis: EMPH_LEVEL;
        -- Emphasis logically applying to the entirety of this entry.
        -- The value of this attribute is propagated to the same-named
        -- attributes in EHCR_INFO and its subclasses.

    recorder: STAFF_MEMBER;
        -- Actual person recording the entry electronically.

    shadow_auth: PERSON;
        -- Void unless a student entry in which case this is
        -- the student responsible for the entry.

invariant
    name /= void;

end -- class EHCR_ENTRY

```

6.19 Class Observation

```
-- item\OBSERVAT.E
-- Version 1.0 1995-6-30
--
-- A deferred class modelling the idea of an observed, measured or
-- factual piece of medical information or factual piece of Health
-- Care information.
--
```

```
class
  OBSERVATION
```

```
inherit
  EHCR_ENTRY
```

```
feature -- Context
```

```
info_provider: PERSON;
  -- The person actually providing the information.
  -- usually an HCP but may be e.g. Patient or
  -- a member of the Patient's family
```

```
cx_comment: MULTI_TEXT;
  -- A comment about the context (ie circumstances) in which
  -- care was delivered.
  -- Useful for qualifying the entries made either
  -- As HRIs or Collections
```

```
feature -- Content
```

```
links: ARRAY [GEHR_UID];
  -- Cross-references to other OBSERVATIONS in the same or another
  -- transaction, or possibly a transaction itself.
```

```
in_reply_to: LIST [OBSERVATION];
  -- The OBSERVATION information resulting from a request.
```

```
access_rights: PERMISSIONS;
  -- the categories of user allowed to see this information
```

```
feature {OBSERVATION} -- Implementation
```

```
annotated_by: SUBSET [HEADING];
  -- the set of headings which annotate
  -- this OBSERVATION.
```

```
end -- class OBSERVATION
```

6.20 Class hri

```
-- item\HRI.E
-- Version 1.0 1995-6-30
--
-- Models a "leaf" observation, ie one which cannot contain other observations.
--
```

```
class
    HRI
```

```
inherit
    OBSERVATION
```

feature

```
content: EHCR_INFO;
    -- An instance of EHCR_INFO (and therefore any subclass).
```

```
ct_comment: MULTI_TEXT;
    -- A comment about the content.
```

```
ct_emphasis: EMPH_LEVEL;
    -- A means of emphasising the content
```

feature

```
dt_observed: DATE_TIME;
    -- Date/time at which the observation was made (cf Date/time
    -- of contact - CONTACT.dt_occurred and Date/time committed
    -- TRANS_VERSION.dt_committed).
```

```
certainty: PLAIN_TEXT;
    -- The degree of certainty attached to an observation or
    -- diagnosis. Records the degree of uncertainty of the
    -- authorising HCP, to make it clear that he/she is not
    -- 100% sure of a recorded diagnosis or observation.
```

```
is_derived: BOOLEAN;
    -- True if the information was derived
    -- by some computation process rather than directly entered
    -- by a person, for example Body Mass Index.
```

```
end -- class HRI
```

6.21 Class hri_collection

```
-- item\HRI_COLL.E
-- Version 1.0 1995-6-30
--
-- This class models a tightly-grouped collection of OBSERVATIONs, which
-- can be HRI_COLLECTIONS and/or HRIs. Instances of this class can be
-- thought of as acting as qualification for member OBSERVATION instances.
-- Th Collection also signifies a change in the scope of the contained
-- data, enabling the data subject to be changed. The scope is always
-- Patient, until a Collection occurs. The scope may refer to other
-- persons whose data may be recorded in this patient's EHCR. The scope
-- will most often be of clinical concepts.
```

class

HRI_COLLECTION

inherit

OBSERVATION

feature -- Content

members: LIST [OBSERVATION];

invariant

not members.empty;

end -- class *HRI_COLLECTION*

6.22 Class heading

```
-- item\HEADING.E
-- Version 1.0 1995-6-30
--
-- Heading.
--
-- This class has the purpose of providing an annotation mechanism for
-- combinations of HRIs and Collections.
```

class

HEADING

inherit

EHCR_ENTRY

feature

parent: like Current;

-- Parent heading. Models nested headings. This attribute

-- can only be used for headings attached to the same OBSERVATION.

end -- class *HEADING*

EHCR_Info Cluster

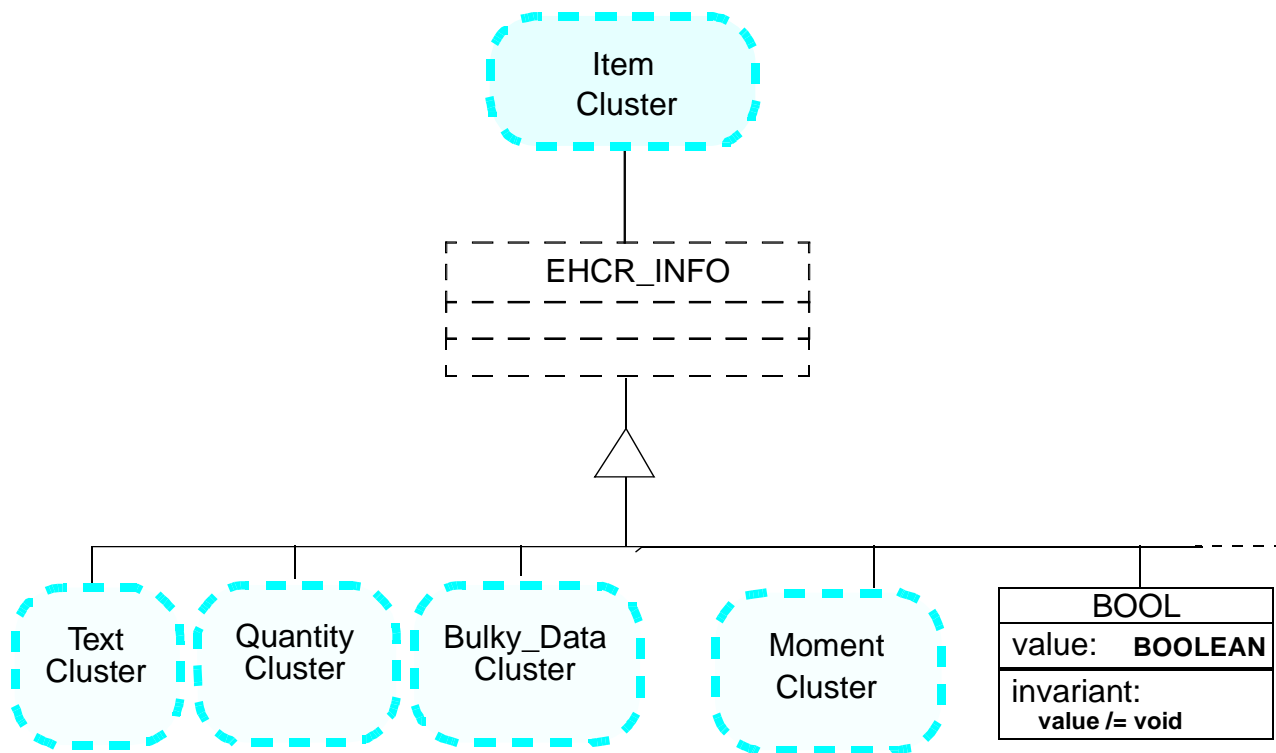


FIGURE 13 EHCR Info Cluster

6.23 Class ehcr_info

```
-- ehcr_inf\EHCR_INF.E
-- Version 1.0 1995-6-30
--
-- Models the content of an HRI of any type
--
```

deferred class

EHCR_INFO

end -- class *EHCR_INFO*

6.24 Class bool

```
-- ehcr_inf\BOOL.E
-- Version 1.0 1995-6-30
--
-- Models an piece of Boolean data.
--
```

class

BOOL

inherit

EHCR_INFO

feature -- Content

value: BOOLEAN;

invariant

value /= void;

end -- class *BOOL*

Text Cluster

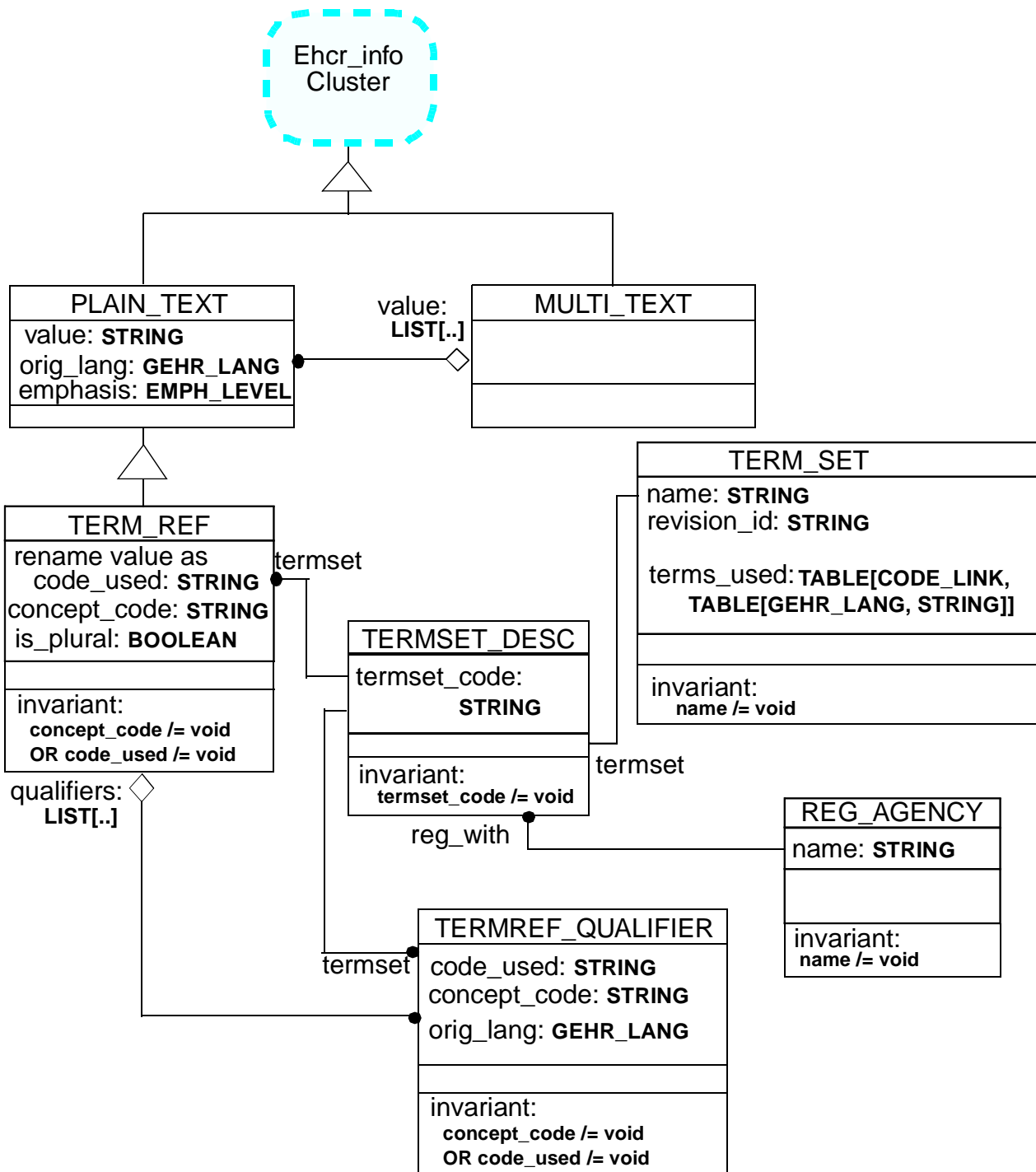


FIGURE 14 Text Cluster

6.25 Class plain_text

```

-- text\PLAIN_TE.E
-- Version 1.0 1995-6-30
--
-- Models plain text. This may be as typed in by a user, or it may be
-- in fact a term, which is plain text derived by more involved means.
-- Not to be confused with WP text containing formatting information,
-- which is a multi-media type. Since the term_ref inherits from
-- PLAIN_TEXT, coded terms can be used wherever a PLAIN_TEXT
-- is specified.
--
class
    PLAIN_TEXT

inherit
    EHCR_INFO

feature -- Access

    value: STRING;
        -- the actual text, represented as a STRING

feature -- Multilingual

    orig_lang: GEHR_LANG;
        -- The language in which this piece of information was
        -- originally recorded. Note that the
        -- names of EHCR_ENTRYs, and also OBSERVATION comments,
        -- will all have original language recorded for them.

feature -- context

    emphasis: EMPH_LEVEL;
        -- To enable emphasis to be added to a piece of text
        -- wherever it is used.

end -- class PLAIN_TEXT

```

6.26 Class multi_text

```

-- text\MULTI_TE.E
-- Version 1.0 1995-6-30
--

```

-- Multiple text items. A MULTI_TEXT instance contains a LIST of
 -- PLAIN_TEXTs, which means that the actual items in the LIST could be
 -- any combination of PLAIN_TEXT and TERM_REF.
 --

class

MULTI_TEXT

inherit

EHCR_INFO

feature --

value: LIST [PLAIN_TEXT];

end -- class *MULTI_TEXT*

6.27 Class term_ref

-- text\TERM_REF.E
 -- Version 1.0 1995-6-30
 --
 -- TERM_REF encapsulates the idea of a code for a term
 --

class

TERM_REF

inherit

PLAIN_TEXT

rename

value as code_used

end

feature

concept_code: STRING;

-- The code for the concept.

-- This is the code for the preferred term if one exists.

termset: TERMSET_DESC;

-- The term set descriptor corresponding to the termset

-- containing this code. This may be a local or

-- a registered termset. Unambiguously identifies the

```
-- termset, local or registered, which should be used to
-- interpret the code.
```

```
is_plural: BOOLEAN;
```

```
-- plurality of the noun implied by the code_value used.
-- This can be used to generate the correct spellings
-- of plural versions of nouns.
```

```
qualifiers: LIST [TERMREF_QUALIFIER];
```

```
-- the additional codes and their termsets which go to
-- make up a compound termref.
```

```
invariant
```

```
code_used /= void or concept_code /= void;
```

```
end -- class TERM_REF
```

6.28 Class termref_qualifier

```
-- text\TERMREF_E
```

```
-- Version 1.0 1995-6-30
```

```
--
```

```
-- To provide a mechanism for adding other terms to a term_ref to
-- form a compound term.
```

```
-- No particular model of healthcare is assumed, and no guarantee
-- given that the compound term makes sense.
```

```
-- This form of aggregation of terms will be useful in naming
```

```
-- Entries in the record.
```

```
-- Another form of aggregation of terms is provided by the Multi_text
```

```
-- class, for use in providing supplementary data about the content
```

```
-- values of Observations.
```

```
class
```

```
TERMREF_QUALIFIER
```

```
feature
```

```
concept_code: STRING;
```

```
-- the code used if a preferred term from a
-- registered term set is used.
```

```
code_used: STRING;
```

```
-- the code used if this is not the preferred term or
-- if a local code is used.
```

termset: TERMSET_DESC;

-- The term set descriptor corresponding to the termset
 -- containing this code. This may correspond to a local or
 -- a registered term set. Unambiguously identifies the
 -- termset, local or registered, which should be used to
 -- interpret the code.

orig_lang: GEHR_LANG;

invariant

concept_code /= void **or** *code_used* /= void;

end -- class *TERMREF_QUALIFIER*

6.29 Class term_set

-- text\TERMSET.E
 -- Version 1.0 1995-6-30
 -- Minimum Model of a Term Set in the Gehr Context.

class

TERM_SET

feature -- Access

name: STRING;

-- the name of the termset e.g. Read, ICD10

revision_id: STRING;

-- the revision of the termset

terms_used: TABLE [CODE_LINK, TABLE [GEHR_LANG, STRING]];

-- This table is meant to contain at least all the codes
 -- and terms used in this EHCR. It is included here to
 -- enable an EHCR_EXTRACT to be modelled without any
 -- special provision for extracting and transmitting
 -- terms used.
 -- An implementation is free to handle its termsets outside
 -- the EHCR provided that suitable extract methods are
 -- available to deliver the terms used in the manner
 -- specified here.

invariant

name /= void;

end -- class *TERM_SET*

6.30 Class *reg_agency*

-- text\REG_AGEN.E
 -- Version 1.0 1995-6-30
 --
 -- Models term set registering agency
 --

class
 REG_AGENCY

feature -- Identification

name: STRING;
 -- Name of the registering agency, eg "CEN", "GEHR".

invariant
 name /= void;

end -- class *REG_AGENCY*

6.31 Class *termset_desc*

-- text\TERMSET_.E
 -- Version 1.3 1995-6-13
 --
 -- This class models the a term set descriptor. A code set has a name,
 -- and may be local or it may be registered with an agency.
 --

class
 TERMSET_DESC

feature -- Access

termset_code: STRING;
 -- the local code for the term set used.

termset: TERM_SET;
 -- the set of terms containing this term

reg_with: REG_AGENCY;

-- Agency with which this code set is registered.

invariant

termset_code /= void;

end -- class *TERMSET_DESC*

Quantity Cluster

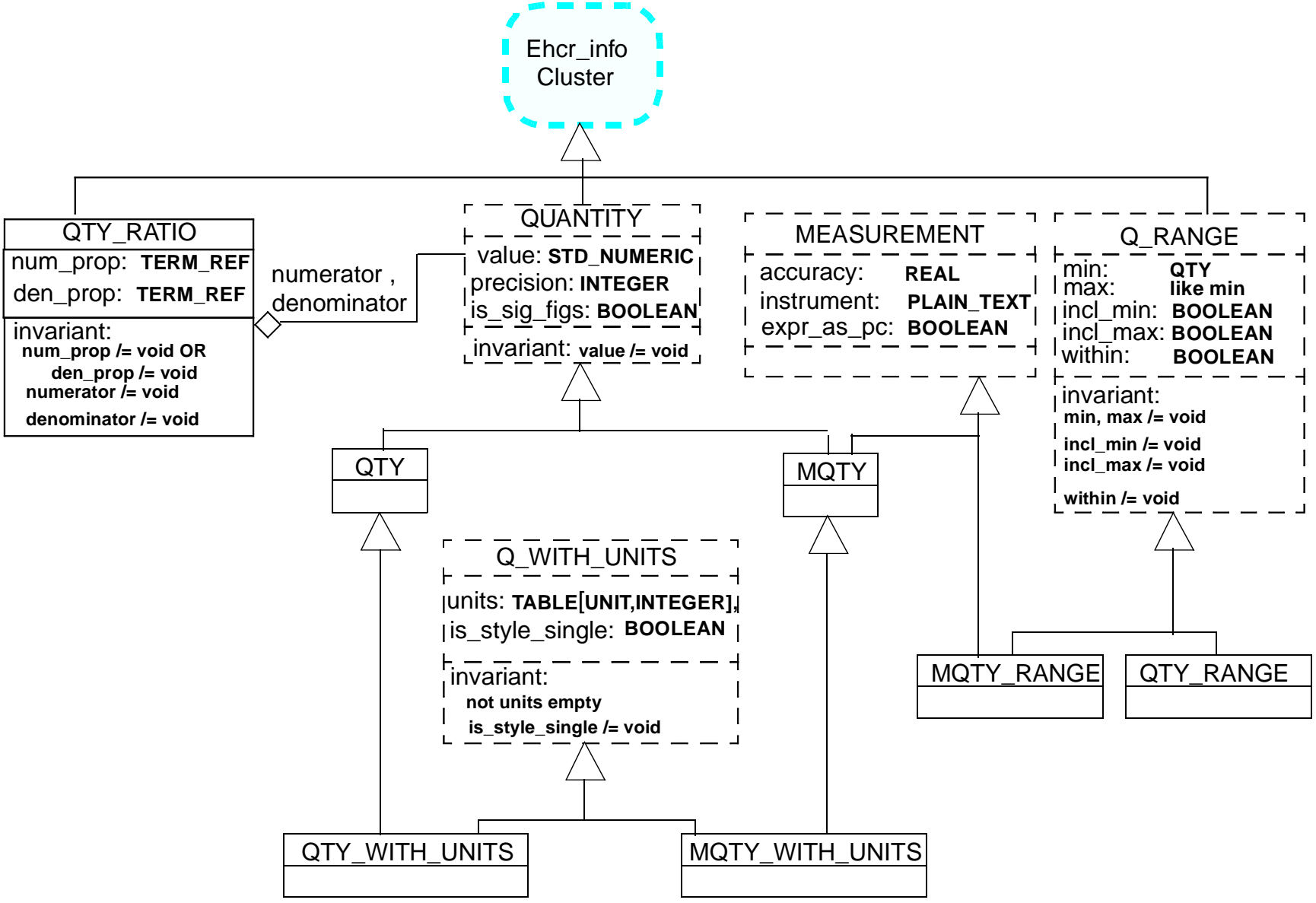


FIGURE 15 Quantity Cluster

6.32 Class measurement

```
-- quantity\MEASUREM.E
-- Version 1.0 1995-6-30
--
-- Models a measured quantity by adding the concept of accuracy
-- and the instrument of measurement.
--
```

```
deferred class
  MEASUREMENT
```

```
feature -- Content
```

```
  accuracy: REAL;
    -- Accuracy to which the value of the quantity is expressed.
```

```
  instrument: PLAIN_TEXT;
    -- Name of instrument with which measurement was made.
    -- Value is the name of an actual instrument, or values such
    -- as "by eye", "manually" etc are acceptable.
```

```
  expr_as_pc: BOOLEAN;
    -- True if the accuracy was expressed as a percentage
    -- rather than in the same units as value.
```

```
end -- class MEASUREMENT
```

6.33 Class mqty

```
-- quantity\MQTY.E
-- Version 1.0 1995-6-30
--
-- Models a quantity with value and measurement.
--
```

```
class
  MQTY
```

```
inherit
  QUANTITY;
  MEASUREMENT
```

```
end -- class MQTY
```


6.34 Class mqty_range

```
-- quantity\MQTY_RANGE
-- Version 1.0 1995-6-30
--
-- Models a quantity range that is measured.
```

class

MQTY_RANGE

inherit

Q_RANGE;
MEASUREMENT

end -- class *MQTY_RANGE*

6.35 Class mqty_with_units

```
-- quantity\MQTY_WITH_UNITS
-- Version 1.0 1995-6-30
--
-- Models a measured quantity recorded with units.
--
```

class

MQTY_WITH_UNITS

inherit

MQTY;
Q_WITH_UNITS

end -- class *MQTY_WITH_UNITS*

6.36 Class q_range

```
-- quantity\Q_RANGE.E
-- Version 1.3 1995-6-13
--
-- Models a range. Any range with two limits, and any combination of
-- inclusion of the limit values, as well as the sense of the range
-- (within or without) can be modelled. Single-sided ranges are
-- modelled by setting one limit to the infinity value.
--
```

deferred class

Q_RANGE

inherit

EHCR_INFO

feature -- Content

min: QTY;

-- Limits the defining range. Single-sided ranges are modelled
-- by setting one limit to the infinity value
-- (negative infinity in the case of min)

max: like min;

-- Limits the defining range. Single-sided ranges are modelled
-- by setting one limit to the infinity value
-- (positive infinity in the case of max)

incl_min: BOOLEAN;

-- Indicates whether the range limit values themselves
-- are included. Effectively models the difference
-- between "<" and "<=".

incl_max: BOOLEAN;

-- Indicates whether the range limit values themselves
-- are included. Effectively models the difference
-- between ">" and ">=".

within: BOOLEAN;

-- Indicates whether the range is inside or outside the limits

invariant

min /= void and max /= void and incl_min /= void and incl_max /= void and within /= void;

end -- class *Q_RANGE*

6.37 Class q_with_units

-- quantity\Q_WITH_U.E

-- Version 1.0 1995-6-30

--

-- Models the basic properties of quantified entities by defining
-- units with which they are expressed.

--

deferred class

*Q_WITH_UNITS**feature**units: TABLE [UNIT, INTEGER];**-- units in which the quantity was recorded**-- the INTEGER is the exponent**is_style_single: BOOLEAN;**-- True for "15.27 kg" style, False for "12st 10lb 7.3oz" style**invariant**not units.empty;**is_style_single /= void;**end -- class Q_WITH_UNITS*

6.38 Class qty_range

*-- quantity\QTY.E**-- Version 1.0 1995-6-30**--**-- Models a quantity with value but no measurement aspects.**--**class**QTY**inherit**QUANTITY**end -- class QTY**-- quantity\QTY_RANG.E**-- Version 1.0 1995-6-30**--**-- Models a quantity range that does not have measurement aspects.**class**QTY_RANGE**inherit**Q_RANGE*

end -- class *QTY_RANGE*

6.39 Class qty_ratio

-- quantity\QTY_RATIO.E
 -- Version 1.0 1995-6-30
 --
 -- Models a ratio of two quantities such as "250 mg/500 ml". It is
 -- also possible to model e.g. "250 mg solute per 500 ml solvent".
 --

class
 QTY_RATIO

inherit
 EHCR_INFO

feature -- Content

numerator: QUANTITY;
 -- Numerator and denominator of the ratio.

denominator: QUANTITY;
 -- Numerator and denominator of the ratio.

num_prop: TERM_REF;
 -- What is being quantified in the numerator and
 -- denominator.

den_prop: TERM_REF;
 -- What is being quantified in the numerator and
 -- denominator.

invariant
 (*num_prop /= void or den_prop /= void*) **and** *numerator /= void and denominator /= void;*

end -- class *QTY_RATIO*

6.40 Class qty_with_units

-- quantity\QTY_WITH.E
 -- Version 1.0 1995-6-30
 --
 -- Models a quantity recorded with units.
 --

```

class
    QTY_WITH_UNITS

inherit
    QTY;
    Q_WITH_UNITS

end -- class QTY_WITH_UNITS

```

6.41 Class quantity

```

-- quantity\QUANTITY.E
-- Version 1.0 1995-6-30
--
-- Models a quantity with value and optional units and optional measurement.
--

class
    QUANTITY

inherit
    EHCR_INFO

feature -- Content

    value: STD_NUMERIC;
        -- The actual value of the quantity.

    precision: INTEGER;
        -- Precision to which the value of the quantity is
        -- expressed. The precision is expressed as an integer,
        -- meaning the number of significant figures or decimal
        -- places. This may be void if not known.

    is_sig_figs: BOOLEAN;
        -- whether precision indicates a number of decimal places
        -- or significant figures.
        -- This may be void if precision is void

invariant
    value /= void;

end -- class QUANTITY

```

Units Cluster

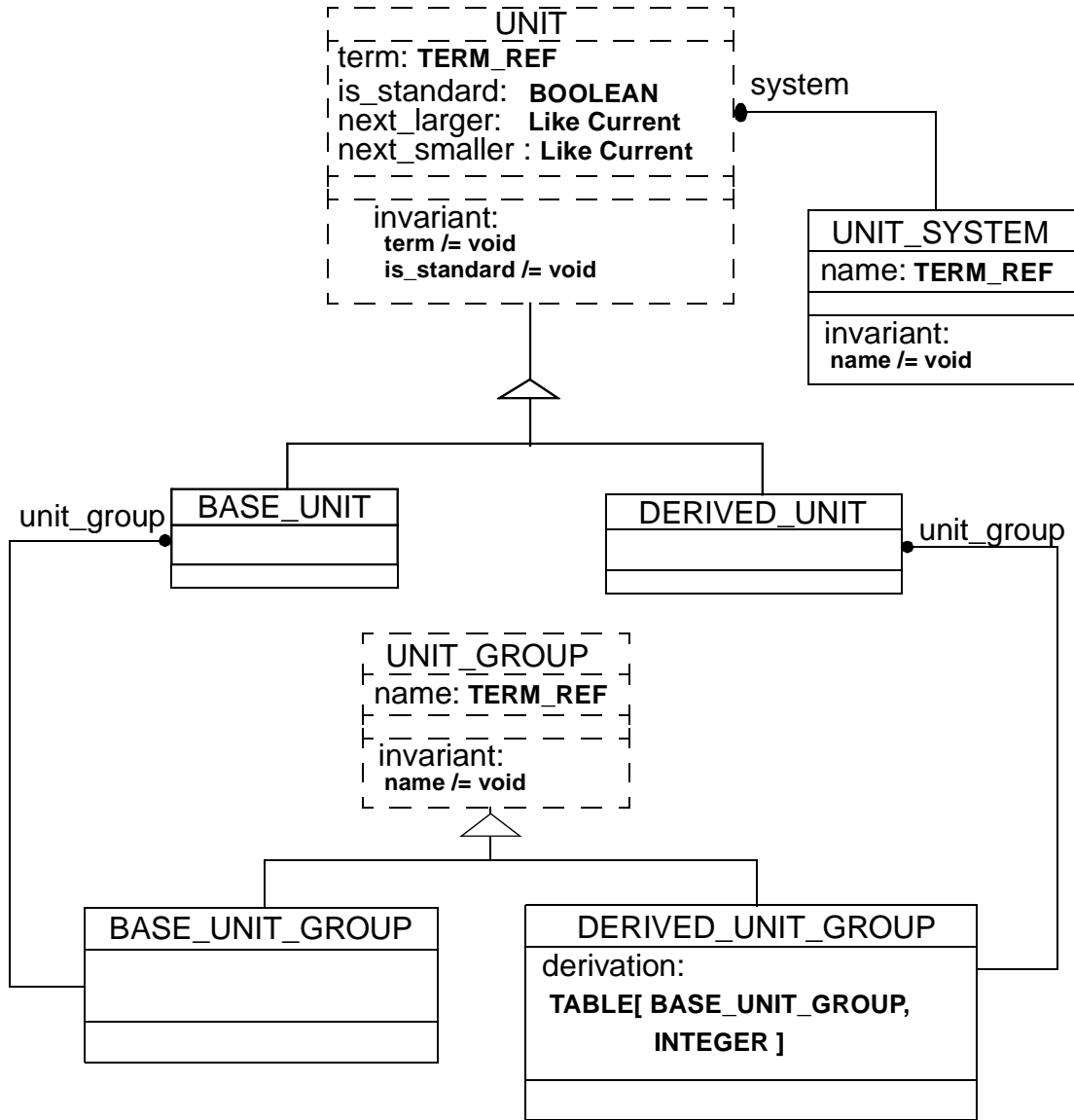


FIGURE 16 Units Cluster

6.42 Class base_unit

```
-- units\BASE_UNI.E
-- Version 1.0 1995-6-30
--
-- Models a Base Unit or Supplementary Unit.
--

class
    BASE_UNIT

inherit
    UNIT

feature

    unit_group: BASE_UG;
        -- the group of units to which this unit belongs
        -- e.g. mass, length ...

end -- class BASE_UNIT
```

6.43 Class base_unit_group

```
-- units\BASE_UG.E
-- Version 1.0 1995-6-30
--
-- Models groups of Base Units e.g. mass, length ...
--

class
    BASE_UG

inherit
    UNIT_GROUP

end -- class BASE_UG
```

6.44 Class derived_unit

```
-- units\DERVD_UN.E
-- Version 1.0 1995-6-30
--
-- Models a Derived Unit.
```

--

*class**DERIVED_UNIT**inherit**UNIT**feature**unit_group: DERVD_UG;*

-- the group of units to which this unit belongs

-- e.g. pressure, volume ...

end -- class *DERIVED_UNIT*

6.45 Class derived_unit_group

-- units\DERVD_UG.E

-- Version 1.0 1995-6-30

--

-- Models groups of Derived Units e.g. pressure, volume, ...

--

*class**DERIVED_UNIT_GROUP**inherit**UNIT_GROUP**feature**derivation: TABLE [BASE_UG, INTEGER];*

-- base unit groups of which the derived unit group is composed

-- e.g. volume = (length, 3)

end -- class *DERIVED_UNIT_GROUP*

6.46 Class unit

-- units\UNIT.E

-- Version 1.0 1995-6-30

--

-- Models a recognised Unit e.g. kg, minutes, ...

--

deferred class*UNIT***feature***term: TERM_REF;*

-- the term for the unit e.g. metre, second, Pascal, Litre

feature*is_standard: BOOLEAN;*

-- whether the unit is normally shown. False for e.g. deciLitres

next_larger: like Current;

-- the next larger unit in a hierarchy.

next_smaller: like Current;

-- the next smaller unit in a hierarchy.

system: UNIT_SYSTEM;

-- the unit system to which this unit belongs.

invariant*term /= void;**is_standard /= void;***end** -- class *UNIT*

6.47 Class *unit_group*

-- units\UNIT_GRO.E

-- Version 1.0 1995-6-30

--

-- Models groups of Units e.g. mass, pressure, ...

--

deferred class*UNIT_GROUP***feature***name: TERM_REF;*

-- the name of the unit group e.g. mass, pressure

invariant

name /= void;

end -- class *UNIT_GROUP*

6.48 Class unit_system

-- units\UNIT_SYS.E

-- Version 1.0 1995-6-30

--

-- Provides the minimum model of a Unit system.

--

class

UNIT_SYSTEM

feature -- Identification

name: TERM_REF;

-- original unit system in which this quantity was

-- recorded e.g. SI, US, Imperial

invariant

name /= void;

end -- class *UNIT_SYSTEM*

Bulky Data Cluster

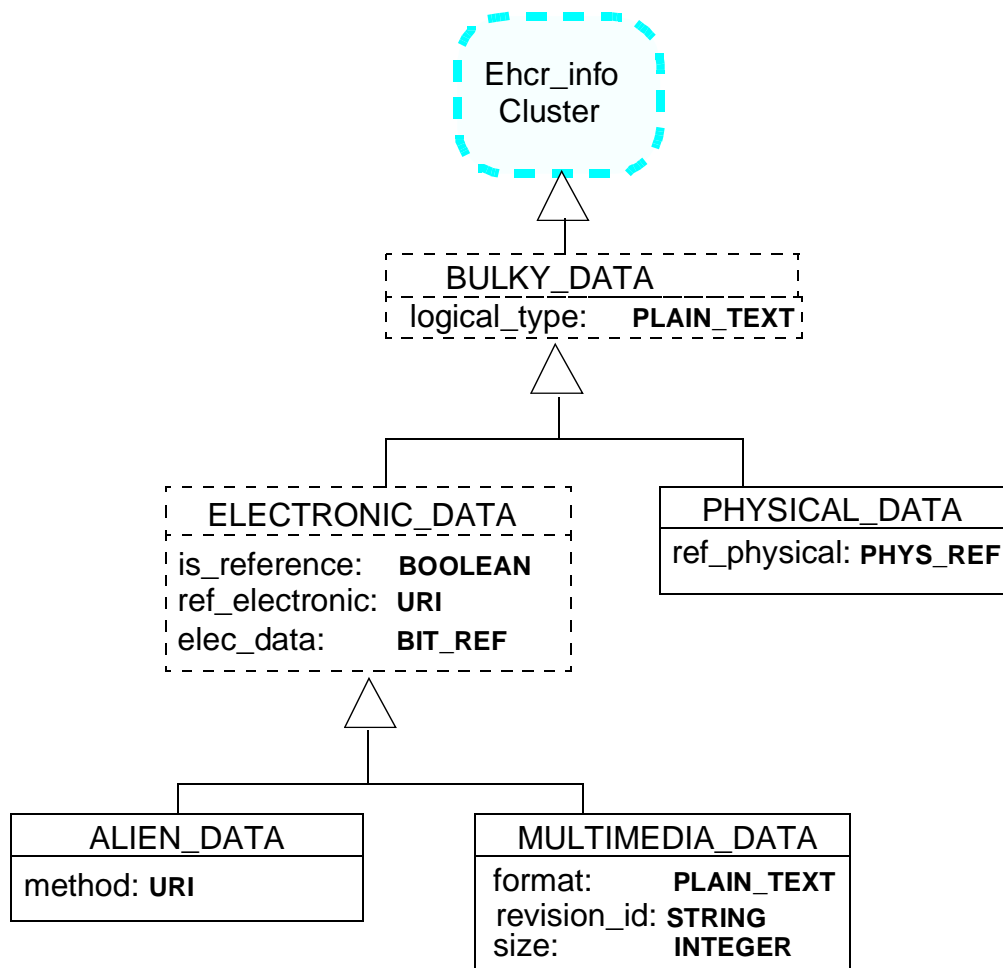


FIGURE 17 Bulky Data Cluster

6.49 Class bulky_data

```
-- bulky\BULKY_DA.E
-- Version 1.0 1995-6-30
--
-- Model of an opaque piece of bulky data.
--
```

```
deferred class
    BULKY_DATA
```

```
inherit
    EHCR_INFO
```

```
feature -- identification
```

```
    logical_type: PLAIN_TEXT;
        -- Logical type, eg "drawing", "image", "graph" etc
```

```
end -- class BULKY_DATA
```

6.50 Class electronic_data

```
-- bulky\ELECTRON.E
-- Version 1.0 1995-6-30
--
-- Model of a piece of bulky data stored in electronic format.
--
```

```
deferred class
    ELECTRONIC_DATA
```

```
inherit
    BULKY_DATA
```

```
feature -- Identification
```

```
    is_reference: BOOLEAN;
        -- True if the data is not stored in expanded form,
        -- ie within the EHCR itself.
```

```
    ref_electronic: URI;
        -- URI reference to electronic information
```

feature

```
elec_data: BIT_REF;  
    -- the actual data
```

```
end -- class ELECTRONIC_DATA
```

6.51 Class alien_data

```
-- bulky\ALIEN_DA.E  
-- Version 1.0 1995-6-30  
--  
-- Model of a piece of "alien" information, i.e. a lump of data about which  
-- nothing is known outside of its origin.  
--
```

```
class  
    ALIEN_DATA
```

```
inherit  
    ELECTRONIC_DATA
```

```
feature -- access
```

```
method: URI;  
    -- the location of the software to interpret these data
```

```
end -- class ALIEN_DATA
```

6.52 Class multimedia_data

```
-- bulky\MULTIMED.E  
-- Version 1.0 1995-6-30  
--  
-- Model of an opaque piece of multi-media information.  
--
```

```
class  
    MULTIMEDIA_DATA
```

```
inherit  
    ELECTRONIC_DATA
```

```
feature -- Identification
```

format: PLAIN_TEXT;
-- mm_data format, eg JPEG, GIF, WP51, SGML etc

revision_id: STRING;
-- revision of the format used

feature

size: INTEGER;
-- Number of bytes.

end -- class *MULTIMEDIA_DATA*

6.53 Class `physical_data`

```
-- bulky\PHYSICAL.E
-- Version 1.0 1995-6-30
--
-- Model of a piece of bulky data which is a reference to some physical data..
--

class
    PHYSICAL_DATA

inherit
    BULKY_DATA

feature -- Identification

    ref_physical: PHYS_REF;
        -- reference to physical object

end -- class PHYSICAL_DATA
```

Moment Cluster

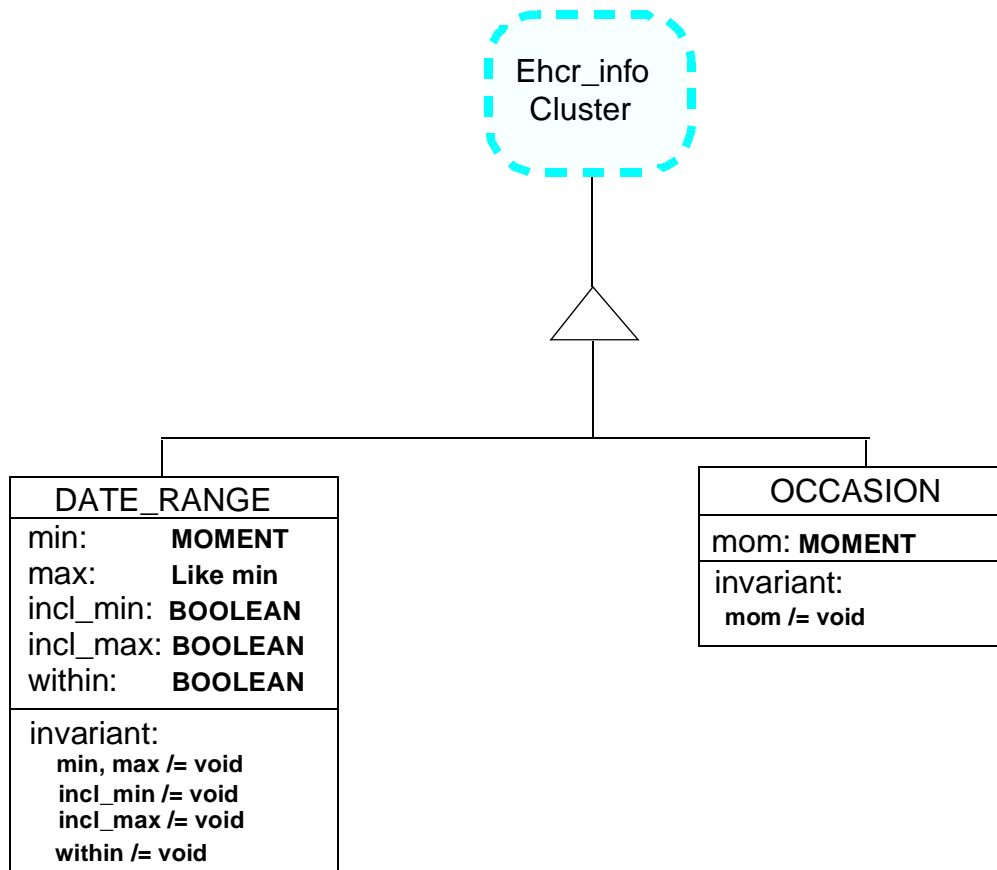


FIGURE 18 Moment Cluster

6.54 Class date_range

```

-- moment\DATE_RANGE
-- Version 1.0 1995-6-30
--
-- Models a Range of dates, times, or dates+times
--

class
    DATE_RANGE

inherit
    EHCR_INFO

feature -- Content

    min: MOMENT;
        -- Limits the defining range. Single-sided ranges are modelled
        -- by setting one limit to the infinity value
        -- (negative infinity in the case of min)

    max: like min;
        -- Limits the defining range. Single-sided ranges are modelled
        -- by setting one limit to the infinity value
        -- (positive infinity in the case of max)

    incl_min: BOOLEAN;
        -- Indicates whether the range limit values themselves
        -- are included. Effectively models the difference
        -- between "<" and "<=".

    incl_max: BOOLEAN;
        -- Indicates whether the range limit values themselves
        -- are included. Effectively models the difference
        -- between ">" and ">=".

    within: BOOLEAN;
        -- Indicates whether the range is inside or outside the limits

invariant
    min /= void and max /= void and incl_min /= void and incl_max /= void and within /=
void;

end -- class DATE_RANGE

```

6.55 Class occasion

-- moment\OCCASION.E
-- Version 1.0 1995-6-30
-- to represent a single occasion as the content of an HRI

class

OCCASION

inherit

EHCR_INFO

feature -- Content

mom: MOMENT;

invariant

mom /= void;

end -- class *OCCASION*

Persons and Places Cluster

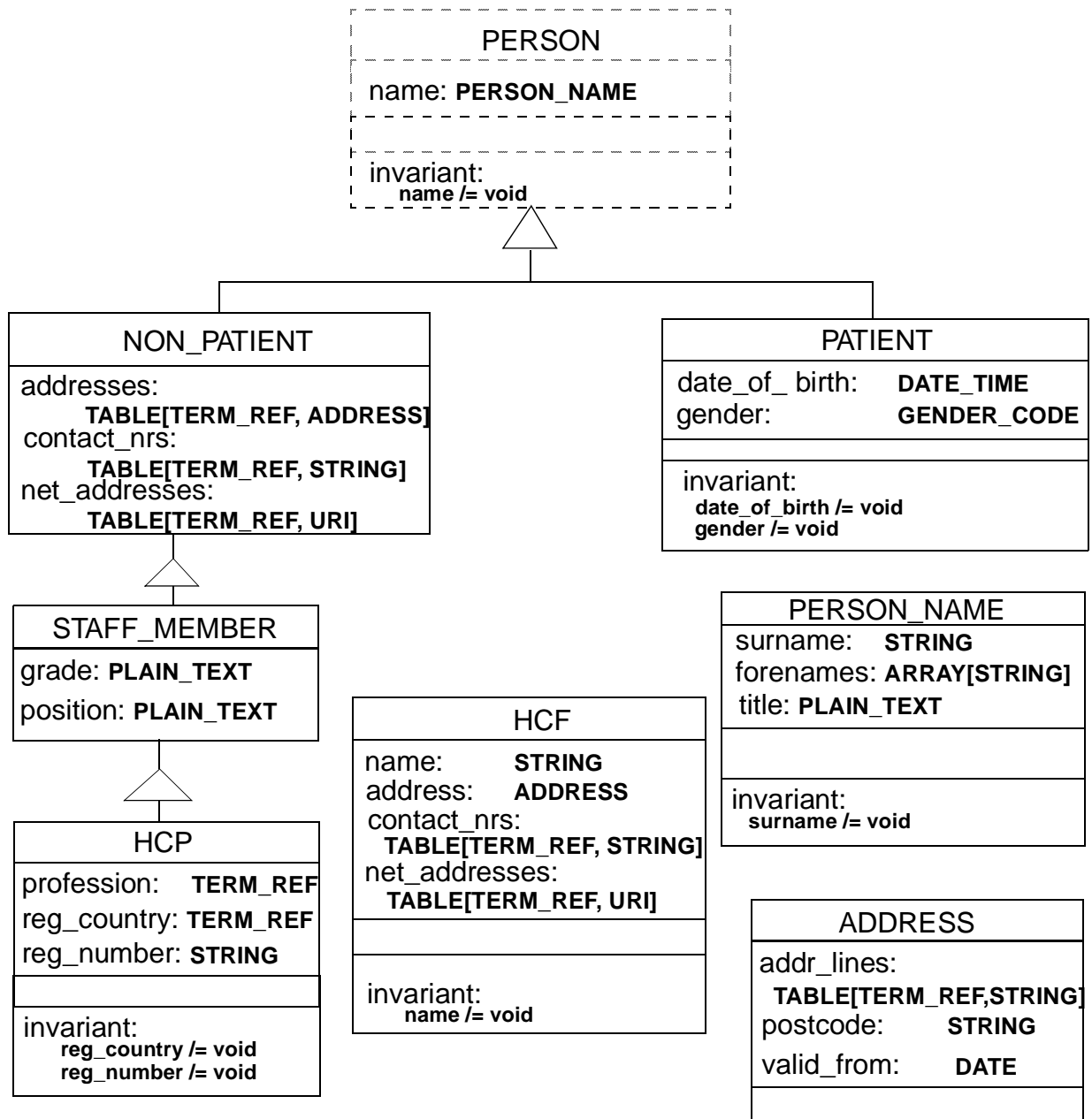


FIGURE 19 Persons and Places Cluster

6.56 Class person

```
-- people\PERSON.E
-- Version 1.0 1995-6-30
--
-- Basic model of a person, containing the absolute minimum details
-- needed for GEHR compliance.
--
```

```
deferred class
    PERSON
```

```
feature --
```

```
    name: PERSON_NAME;
```

```
invariant
```

```
    name /= void;
```

```
end -- class PERSON
```

6.57 Class non_patient

```
-- people\NON_PATLE
-- Version 1.0 1995-6-30
--
-- Basic model of a person other than the Patient, containing the
-- absolute minimum details needed for GEHR compliance.
--
```

```
class
    NON_PATIENT
```

```
inherit
    PERSON
```

```
feature --
```

```
    addresses: TABLE [TERM_REF, ADDRESS];
```

```
    contact_nrs: TABLE [TERM_REF, STRING];
```

```
    net_addresses: TABLE [TERM_REF, URI];
```

end -- class *NON_PATIENT*

6.58 Class patient

-- people\PATIENT.E
-- Version 1.0 1995-6-30
--
-- Model of a PATIENT
--

class
 PATIENT

inherit
 PERSON

feature

date_of_birth: DATE_TIME;

gender: GENDER_CODE;

invariant

date_of_birth /= void;

gender /= void;

end -- class *PATIENT*

6.59 Class staff_member

-- people\STAFF_ME.E
-- Version 1.0 1995-6-30
--
-- Model of a member of staff. This is to differentiate non-professional
-- members of staff from professional staff members. This specialisation
-- of PERSON includes attributes not found in the generic PERSON class.
--

class
 STAFF_MEMBER

inherit
 NON_PATIENT

feature --

grade: PLAIN_TEXT;

position: PLAIN_TEXT;

end -- class *STAFF_MEMBER*

6.60 Class person_name

-- people\PERSON_N.E

-- Version 1.0 1995-6-30

--

-- The name of a person

--

class

PERSON_NAME

feature

surname: STRING;

forenames: ARRAY [STRING];

title: PLAIN_TEXT;

invariant

surname /= void;

end -- class *PERSON_NAME*

6.61 Class address

-- people\ADDRESS.E

-- Version 1.0 1995-6-30

-- Minimum model of an address.

--

class

ADDRESS

feature

```

    addr_lines: TABLE [TERM_REF, STRING];
        -- Any number of lines of text, each denoted by a TERM_REF
        -- typically addrline1, addrline2, town/city, county/state,
        -- country

    postcode: STRING;

    valid_from: DATE;

end -- class ADDRESS

```

6.62 Class *hcp*

```

-- people\HCP.E
-- Version 1.0 1995-6-30
--
-- Model of a Health Care Professional, from the GEHR point of view.
-- Note: this does not seek to replace/prescribe etc a (sophisticated)
-- model of HCPs as may exist in an external database at the HCF -
-- it merely models the minimum semantics needed to make other
-- information in the GEHR model sensible.
--

class
    HCP

inherit
    STAFF_MEMBER

feature

    profession: TERM_REF;
        -- professional area of work, eg "neuro-physiologist"

    reg_country: TERM_REF;
        -- country of professional registration

    reg_number: STRING;
        -- registered under number

invariant
    reg_number /= void;
    reg_country /= void;

end -- class HCP

```

6.63 Class hcf

```
-- people\HCF.E
-- Version 1.0 1995-6-30
--
-- Models a Health Care Facility.
--
```

class

HCF

feature -- Identification

name: STRING;

address: ADDRESS;

contact_nrs: TABLE [TERM_REF, STRING];

net_addressees: TABLE [TERM_REF, URI];

invariant

name /= void;

end -- class *HCF*

Basic Cluster

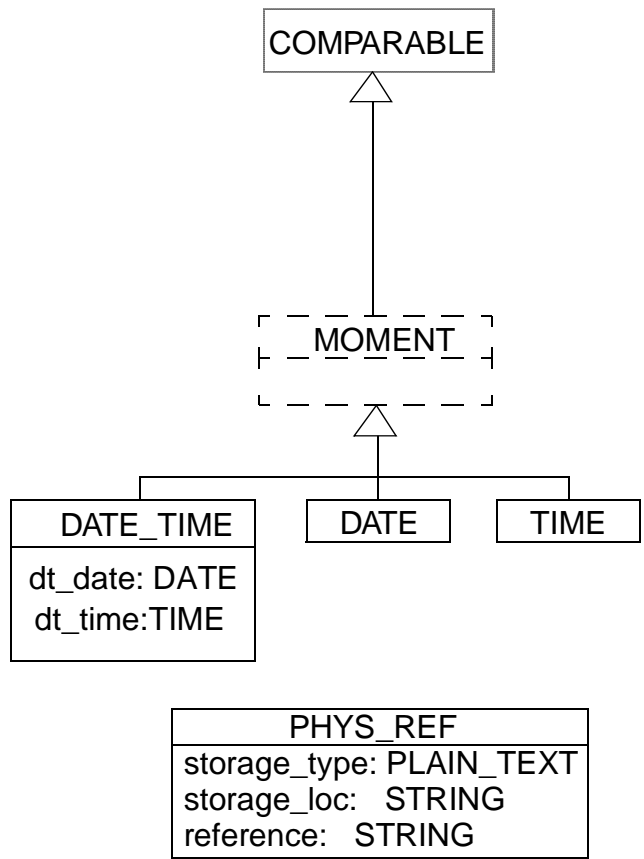


FIGURE 20 GEHR Basic Cluster

6.64 Class date_time

-- basic\DATE_TIME.E
-- Version 1.0 1995-6-30

class

DATE_TIME

inherit

MOMENT

feature

dt_date: DATE;

dt_time: TIME;

end -- class *DATE_TIME*

6.65 Class date

-- basic\DATE.E
-- Version 1.0 1995-6-30

class

DATE

inherit

MOMENT

end -- class *DATE*

6.66 Class time

-- basic\TIME.E
-- Version 1.0 1995-6-30

class

TIME

inherit

MOMENT

end -- class *TIME*

6.67 Class moment

```
-- basic\MOMENT.E
-- Version 1.0 1995-6-30
```

```
deferred class
    MOMENT
```

```
inherit
    COMPARABLE
```

```
end -- class MOMENT
```

6.68 Class phys_ref

```
-- basic\PHYS_REF.E
-- Version 1.0 1995-6-30
--
-- Model of a reference to a physical object. Intended to refer
-- to such things as urine, blood samples, slides, microbiol.
-- materials, and any other physical medical "information".
--
```

```
class
    PHYS_REF
```

```
feature --
```

```
    storage_type: PLAIN_TEXT;
        -- This attribute is a guide to the type of storage being
        -- used. It is intended for information purposes, and is
        -- not needed to resolve the address. Physical:
        -- possibilities include "shelf", "safe", "cold_store"
```

```
    storage_loc: STRING;
        -- This must unambiguously identify the location with
        -- respect to the context where the reference exists. If
        -- storage is in the same HCF, then the value of this
        -- attribute may be relative to that; if the storage is in
        -- another HCF (ie the data containing the reference has
        -- been transmitted) the location should be globally
        -- unique. Values include eg "east wing lab 5 cold room",
        -- "Royal Marsden Hospital, Sutton, Physics Dept., Cold
        -- room 2"
```

```
    reference: STRING;
```

-- Storage reference of item with respect to the location
-- value. Values: eg for shelf storage (of say x-ray
-- slides) there may be a particular numbering system in
-- use for shelving at the location specified by location
-- requiring a reference such as "shelf 5 box 17 sample
-- 21".

end -- class *PHYS_REF*

Exchange Cluster

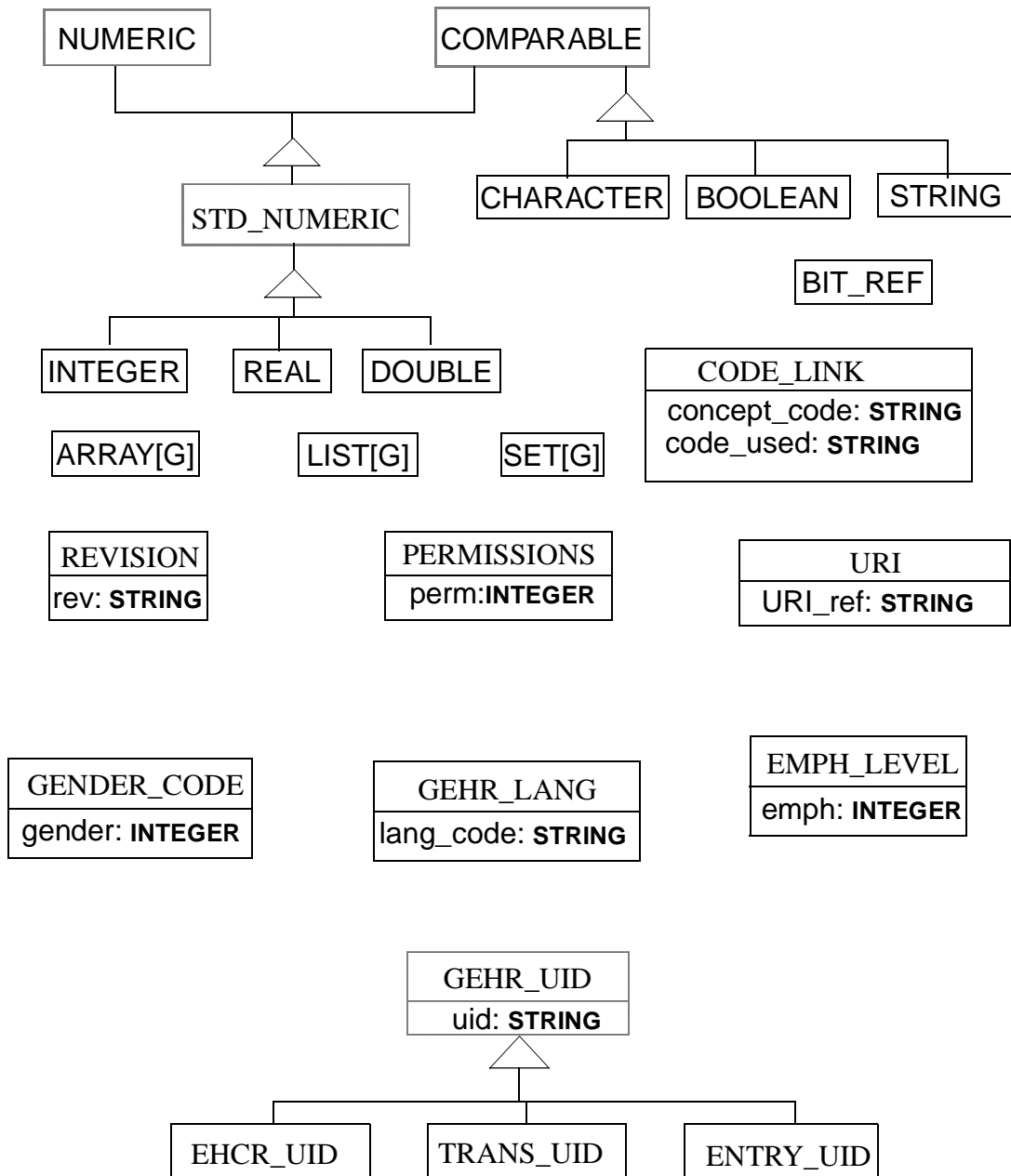


FIGURE 21 GEHR Exchange Cluster

6.69 Class *gehr_lang*

-- basic\GEHR_LAN.E
-- Version 1.0 1995-6-30

class

GEHR_LANG

feature -- content

lang_code: STRING;

invariant

;

end -- class *GEHR_LANG*

6.70 Class *gender_code*

-- exchange\GENDER_C.E
-- Version 1.0 1995-6-30

class

GENDER_CODE

feature -- Access

Male: INTEGER is 1;

Female: INTEGER is 2;

Unknown: INTEGER is 3;

gender: INTEGER;

invariant

gender = male or gender = female or gender = unknown;

end -- class *GENDER_CODE*

6.71 Class std_numeric

-- basic\STD_NUM.E
-- Version 1.0 1995-6-30

deferred class
 STD_NUMERIC

inherit
 NUMERIC

end -- class *STD_NUMERIC*

6.72 Class emph_level

-- exchange\EMPH_LEV.E
-- Version 1.0 1995-6-30

class
 EMPH_LEVEL

feature -- Access

Low: INTEGER is 1;

Medium: INTEGER is 2;

High: INTEGER is 3;

emph: INTEGER;

invariant
 emph = low or emph = medium or emph = high;

end -- class *EMPH_LEVEL*

6.73 Class gehr_uid

-- exchange\GEHR_UID.E
-- Version 1.0 1995-6-30
--
-- GEHR Unique Identifier. Used for identifying instances
-- of several key classes.

--

deferred class
 GEHR_UID

feature -- Identification

uid: STRING;

end -- class *GEHR_UID*

6.74 Class ehcr_uid

-- exchange\EHCR_UID.E
-- Version 1.0 1995-6-30
--
-- GEHR Unique Identifier for an EHCR.
--

class
 EHCR_UID

inherit
 GEHR_UID

end -- class *EHCR_UID*

6.75 Class trans_uid

-- exchange\TRANS_UID.E
-- Version 1.0 1995-6-30
--
-- GEHR Unique Identifier for a Transaction
--

class
 TRANS_UID

inherit
 GEHR_UID

end -- class *TRANS_UID*

6.76 Class entry_uid

```
-- exchange\ENTRY_UID.E
-- Version 1.0 1995-6-30
--
-- GEHR Unique Identifier for an OBSERVATION.
--
```

```
class
    ENTRY_UID
```

```
inherit
    GEHR_UID
```

```
end -- class ENTRY_UID
```

6.77 Class permissions

```
-- exchange\PERMISSI.E
-- Version 1.0 1995-6-30
--
-- A simple capabilities model of access permissions.
-- .
-- The capabilities include:
-- .
-- create:create a new version
-- read: read a version
-- delete:remove a version
-- .
-- The possible users include:
-- .
-- the patient
-- the HCP legally responsible for the patient
-- the HCP authorising the current transaction
-- any HCP involved in the patient's care
-- any other staff member
-- any other person
-- .
-- Details of this or another model need to be analysed;
-- different models for this are: capability, stereotype,
--
```

class

PERMISSIONS

feature -- Access

perm: INTEGER;

pat: INTEGER is 1
hcp_leg: INTEGER is 2
hcp_auth: INTEGER is 4
any_hcp: INTEGER is 8
other_staff: INTEGER is 16
other: INTEGER is 32

invariant

perm > 0 and perm < 64;

end -- class *PERMISSIONS*

6.78 Class revision

-- exchange\REVISION.E

-- Version 1.0 1995-6-30

--

-- Revision of a version of a version-controlled entity. This class is

-- where version-naming/numbering rules should be formalised (to be done).

--

class

REVISION

feature -- access

rev: STRING;

-- always 1.0 for first version

end -- class *REVISION*

6.79 Class code_link

```
-- exchange\CODE_LINK.E
-- Version 1.0 1995-6-30
```

class

CODE_LINK

feature

concept_code: STRING;

code_used: STRING;

end -- class *CODE_LINK*

6.80 Class uri

```
-- exchange\URIE
-- Version 1.0 1995-6-30
--
-- Model of a URI, or Universal Resource Identifier, as used on the WWW.
-- See
-- "Universal Resource Identifiers in WWW"
-- Tim Berners-Lee
-- http://info.cern.ch/hypertext/WWW/Addressing/URL/URI\_Overview.html
--
-- This is a World-Wide Web RFC for global identification of
-- resources. In current use on the web, eg by Mosaic and Chimera
-- browsers.
-- Version 1.0 of GEHR sees a URI merely as a string.
--
```

class

URI

feature -- Access

uri_ref: STRING;

end -- class *URI*

7 Communicating Data

This Chapter contains the GEHR Exchange Format (GEF) and describes the conditions for its use.

7.1 Minimum requirements

Only four basic actions need be carried out to transfer EHCR data from one Source to another:

- Construct an EHCR_Extract containing the required transactions,
- Convert the EHCR_Extract to GEF format,
- Send the GEHR Version,
- Send the GEF bit_stream.

This set of actions applies both when the sender is a GEHR-compliant Source of EHCR data and when the sender is GEHR non-compliant, i.e. a “legacy system”.

GEHR does not specify the means of physical transfer. Any scheme agreed between the sender and receiver will suffice provided that the data can be sent and received error-free.

7.2 Extracting data for transmission

7.2.1 GEHR-Compliant systems

An EHCR_Extract must be constructed to contain the required transactions.

At least one ADMIN transaction should be included for confirmation of the patient’s identity.

GEHR recommends that at least the latest ADMIN Transaction be sent.

If local term sets were used, definitions should be included for at least those terms used in the form of a table as defined in the term_set class (see Section 6.29).

Identity data should be included for all persons identified in the Transactions to be sent. See the Persons and Places Cluster classes in Section 6.56 *et seq.*

7.2.2 Legacy Systems

In extracting data from legacy systems, an application program would need to convert its data to the minimum structured form of an EHCR_Extract. Although the entries in the legacy system’s record might just be free text, the data would need to be wrapped up in one or more Transactions in order to be recognizable by the GEHR-compliant receiver. Any persons identified in the record would need to be presented in the form of the classes of the Persons and Places Cluster. Any coded data must be accompanied by the term sets used.

7.3 Receiving data

The main tasks to be performed once the GEHR Version has been received would be typically:

- Confirm that the software can handle the version of GEHR data to be received
- Receive the GEF bit-stream and convert it to the form of an EHCR_Extract for each patient to be received
- Cross check the identity of the patient(s) whose data are contained in the GEF bit_stream
- Decide whether to accept each patient
- For each new patient construct a new EHCR
- Merge the data for selected patients from the GEF with the existing data.

Versioned Transactions received should be merged into the existing patient EHCR using date/time information to assist the process of correct ordering.

Each Versioned Transaction should be committed to the receiver's record by an HCP who becomes the HCP_auth_acq of the Acquired_Versioned_Trans. The attribute was_gehr_source will also be set TRUE or FALSE depending on the GEHR-compliance of the sender.

7.4 The GEHR Exchange Format

The following is a representation of the GEHR Object Model in ASN.1 and represents the first version of the GEHR Exchange Format (GEF). **This is Version 1.0**

Internal identifiers and corresponding pointers have been used where the Model uses one-to-many relationships. The ASN.1 GEF has been compiled to produce C/C++ encoding/decoding routines.

There is no guarantee at the moment that there is a precise mapping between the GEHR Object Model and the GEF, but it is envisaged that future releases of the Architecture the encoding to ASN.1 will have been performed automatically.

Projects using the GEHR Architecture will test the use of the GEF extensively.

GEF DEFINITIONS	::=	
BEGIN		
REVISION	::=	OCTET STRING
PERMISSIONS	::=	INTEGER
URI	::=	OCTET STRING
GEHR-UID	::=	OCTET STRING
TRANS-UID	::=	GEHR-UID
OBS-UID	::=	GEHR-UID
EHCR-UID	::=	GEHR-UID
GENDER-CODE	::=	INTEGER
CHANGE-TYPE	::=	INTEGER
EMPH-LEVEL	::=	INTEGER
GEHR-LANG	::=	OCTET STRING
DATE	::=	OCTET STRING (SIZE (6))
TIME	::=	OCTET STRING (SIZE (6))
STD-NUMERIC	::=	CHOICE {REAL, INTEGER }
BIT-REF	::=	BIT STRING
POINTERTOITEMS	::=	CHOICE { [0] OBSERVATION-ID, [1] HRI-COLLECTION-ID, [2] HRI-ID }
POINTERTOSTAFF-MEMBER	::=	CHOICE { [0] STAFF-MEMBER-ID, [1] HCP-ID }
POINTERTOHCPC	::=	POINTER
POINTERTOHCFC	::=	POINTER
POINTERTOEHCRSOURCE	::=	POINTER

POINTERTOPERSON	::=	CHOICE{ [0] PERSON-ID, [1] NON-PATIENT-ID, [2] PATIENT-ID }
POINTERTOPATIENT	::=	POINTER
POINTERTOHEADING	::=	POINTER
POINTERTOTERMSET-DESC	::=	POINTER
POINTERTOREG-AGENCY	::=	POINTER
OBSERVATION-ID	::=	ID
EHCR-ID	::=	ID
HRI-COLLECTION-ID	::=	ID
HRI-ID	::=	ID
STAFF-MEMBER-ID	::=	ID
HCP-ID	::=	ID
HCF-ID	::=	ID
PERSON-ID	::=	ID
NON-PATIENT-ID	::=	ID
PATIENT-ID	::=	ID
TERMSET-DESC-ID	::=	ID
REG-AGENCY-ID	::=	ID
POINTER	::=	INTEGER
ID	::=	INTEGER
EHCR-SOURCE	::= { name net-addr owning-hcf ehcrs	[APPLICATION 0] SET [0] OCTET STRING, [1] SEQUENCE {TERM-REF, URI OPTIONAL, [2] POINTERTOHCf, [3] SET OF EHCR OPTIONAL }

EHCR	::= { ehcr-id dt-creation hcp-created-by transactions	[APPLICATION 1] SET [0] EHCR-UID, [1] DATE-TIME OPTIONAL, [2] POINTERTOHCPC OPTIONAL, [3] SET OF VERSIONED-TRANS }
EHCR-EXTRACT	::= { EHCR }	[APPLICATION 2] SET
VERSIONED-TRANS	::= { uid dt-created access-rights amend-rights gehr-version versions	[APPLICATION 3] SET [0] TRANS-UID, [1] DATE-TIME, [2] PERMISSIONS, [3] PERMISSIONS, [4] OCTET STRING, [5] SET OF TRANS-VERSION }
ACQUIRED-VERSIONED-TRANS	::=[APPLICATION 4] SET { VERSIONED-TRANS, source-trans-ref was-gehr-source hcp-auth-acq source	[0] TRANS-UID, [2] BOOLEAN, [3] POINTERTOHCPC, [4] POINTERTOEHCRSOURCE }
TRANS-VERSION	::= { revision-id dt-committed type-of-change items hcp-authorising hcp-legally-resp recorder	[0] SET [0] REVISION, [1] DATE-TIME, [2] CHANGE-TYPE OPTIONAL, [3] POINTERTOITEMS OPTIONAL, [4] POINTERTOHCPC, [5] POINTERTOHCPC OPTIONAL, [6] POINTERTOSTAFF-MEMBER OPTIONAL}
STANDARD-TRANS	::= { TRANS-VERSION }	[1] SET
TRIGGER	::= { TRANS-VERSION }	[2] SET
SUMMARY	::= { STANDARD-TRANS, period	[APPLICATION 5]SEQUENCE DATE-RANGE }
REPORT	::= { STANDARD-TRANS, inreplyto	[APPLICATION 6]SEQUENCE OBS-UID }

CONTCARE	::= { STANDARD-TRANS, period	[APPLICATION 7]SEQUENCE DATE-RANGE }
ADMIN	::= { STANDARD-TRANS, subject	[APPLICATION 8]SEQUENCE POINTERTOPATIENT }
NOTEBENE	::= { STANDARD-TRANS }	[APPLICATION 9]SEQUENCE
CONTACT	::= { STANDARD-TRANS, dtoccurred contact-with	[APPLICATION 10]SEQUENCE DATE-TIME, POINTERTOHCPC }
EHCR-ENTRY	::= { uid name emphasis recorder shadow-auth	[3] SET [0] EHCR-UID, [1] TERM-REF, [2] EMPH-LEVEL OPTIONAL, [3] POINTERTOSTAFF-MEMBER OPTIONAL, [4] POINTERTOPERSON OPTIONAL }
HEADING	::= { EHCR-ENTRY, parent	[APPLICATION 11]SEQUENCE POINTERTOHEADING }
OBSERVATION	::= { links access-rights cx-comment in-reply-to annotated-by	[4] SET [0] EHCR-ENTRY, [1] SET OF GEHR-UID OPTIONAL, [2] PERMISSIONS OPTIONAL, [3] MULTI-TEXT OPTIONAL, [4] SEQUENCE OF OBSERVATION-ID OPTIONAL, [5] SET OF OBSERVATION-ID OPTIONAL }
HRI-COLLECTION	::= { hri-collection-id members	[APPLICATION 12]SET [0] EHCR-ENTRY, [1] HRI-COLLECTION-ID, [2] SET OF OBSERVATION-ID }
HRI	::= {	[APPLICATION 13]SET [0] EHCR-ENTRY,

	hri-id	[1] HRI-ID,	
	ct-comment	[2] MULTI-TEXT	OPTIONAL,
	dt-observed	[3] DATE-TIME	OPTIONAL,
	certainty	[4] PLAIN-TEXT	OPTIONAL,
	is-derived	[5] BOOLEAN	OPTIONAL,
	ct-emphasis	[6] EMPH-LEVEL	OPTIONAL }
EHCR-INFO	::= {	[5] SET	}
PLAIN-TEXT	::= { value orig-lang emphasis	[APPLICATION 14]SET [0] EHCR-INFO, [1] OCTET STRING, [2] GEHR-LANG [3] EMPH-LEVEL	OPTIONAL, OPTIONAL }
MULTI-TEXT	::= { value	[APPLICATION 15]SEQUENCE SEQUENCE OF PLAIN-TEXT }	
TERM-REF	::= { code-used concept-code isplural qualifiers termset	[APPLICATION 16]SET [0] PLAIN-TEXT, [1] OCTET STRING [2] OCTET STRING [3] BOOLEAN [4] SEQUENCE OF TERMREF-QUALIFIER [5] POINTERTOTERMSET-DESC	OPTIONAL, OPTIONAL, OPTIONAL, OPTIONAL, OPTIONAL }
TERMSET-DESC	::= { termset-desc-id termset-code reg-with OPTIONAL, termset	[APPLICATION 17]SET [0] TERMSET-DESC-ID, [1] OCTET STRING, [2] POINTERTOREG-AGENCY [3] TERM-SET	OPTIONAL }
TERM-SET	::= { name revision-id terms-used { CODE-LINK, SEQUENCE { GEHR-LANG, OCTET STRING } }	[APPLICATION 18]SET [0] OCTET STRING, [1] OCTET STRING [2] SEQUENCE	OPTIONAL, OPTIONAL }
REG-AGENCY	::= { reg-agency-id name	[APPLICATION 19]SET [0] REG-AGENCY-ID, [1] OCTET STRING }	

TERMREF-QUALIFIER	::= { code-used concept-code orig-lang termset	[APPLICATION 20]SET [0] OCTET STRING OPTIONAL, [1] OCTET STRING OPTIONAL, [2] GEHR-LANG OPTIONAL, [3] TERMSET-DESC OPTIONAL }
QTY-RATIO	::= { num-prop den-prop numerator denominator	[APPLICATION 21]SET [0] EHCR-INFO, [1] TERM-REF OPTIONAL, [2] TERM-REF OPTIONAL, [3] QUANTITY, [4] QUANTITY }
QUANTITY	::= { value precision is-sig-figs	[6] SET [0] EHCR-INFO, [1] STD-NUMERIC, [2] INTEGER OPTIONAL, [3] BOOLEAN OPTIONAL }
Q-RANGE	::= { min max incl-min incl-max within	[7] SET [0] EHCR-INFO, [1] QTY, [2] QTY, [3] BOOLEAN, [4] BOOLEAN, [5] BOOLEAN }
MEASUREMENT	::= { accuracy instrument expr-as-pc	[8] SET [0] REAL OPTIONAL, [1] PLAIN-TEXT OPTIONAL, [2] BOOLEAN OPTIONAL }
QTY	::= {	[APPLICATION 22]SET [0] QUANTITY }
MQTY	::= {	[APPLICATION 23]SET [0] QUANTITY }
Q-WITH-UNITS	::= { units is-style-single	[9] SET [0] SEQUENCE {UNIT, INTEGER }, [1] BOOLEAN }
QTY-WITH-UNITS	::= { QTY,	[APPLICATION 24]SET
Q-WITH-UNITS }		
MQTY-WITH-UNITS ::=	[APPLICATION 25]	SET

	{ MQTY,	
Q-WITH-UNITS }		
MQTY-RANGE	::= { MEASUREMENT,	[APPLICATION 26]SET
Q-RANGE }		
QTY-RANGE	::= { Q-RANGE }	[APPLICATION 27]SET
BULKY-DATA	::= { logical-type	[10] SET [0] PLAIN-TEXT }
ELECTRONIC-DATA ::=	[11] { is-reference ref-electronic elec-data	SET [0] BULKY-DATA, [1] BOOLEAN OPTIONAL, [2] URI OPTIONAL, [3] BIT-REF OPTIONAL }
PHYSICAL-DATA	::= { ref-physical	[APPLICATION 28]SET [0] BULKY-DATA, [1] PHYS-REF }
ALIEN-DATA	::= { method	[APPLICATION 29]SET [0] ELECTRONIC-DATA, [1] URI OPTIONAL }
MULTIMEDIA-DATA ::=	[APPLICATION 30] { format revision-id size	SET [0] ELECTRONIC-DATA, [1] PLAIN-TEXT OPTIONAL, [2] OCTET STRING OPTIONAL, [3] INTEGER OPTIONAL }
DATE-RANGE	::= { min max incl-min incl-max within	[APPLICATION 31]SET [0] EHCR-INFO, [1] MOMENT, [2] MOMENT, [3] BOOLEAN, [4] BOOLEAN, [5] BOOLEAN }
OCCASION	::= { mom	[APPLICATION 32] SET [0] EHCR-INFO, [1] MOMENT }
BOOL	::= {	[APPLICATION 33]SET [0] EHCR-INFO,

	value	[1] BOOLEAN }
PERSON	::= { personid name	[12] SET [0] PERSON-ID, [1] PERSON-NAME }
NON-PATIENT	::= { addresses	[APPLICATION 34]SET [0] PERSON, [1] SET {TERM-REF, ADDRESS } OPTIONAL,
	contact-nrs	[2] SET {TERM-REF, OCTET STRING } OPTIONAL,
	net-addresses	[3] SET {TERM-REF, URI } OPTIONAL }
STAFF-MEMBER	::= { staff-member-id grade position	[APPLICATION 35]SET [0] NON-PATIENT, [1] STAFF-MEMBER-ID, [2] PLAIN-TEXT OPTIONAL, [3] PLAIN-TEXT OPTIONAL }
HCP	::= { hcp-id profession reg-country reg-number	[APPLICATION 36]SET [0] STAFF-MEMBER, [1] HCP-ID, [2] TERM-REF OPTIONAL, [3] TERM-REF OPTIONAL, [4] OCTET STRING }
PATIENT	::= { patient-id date-of-birth gender	[APPLICATION 37]SET [0] PERSON, [1] PATIENT-ID, [2] DATE-TIME, [3] GENDER-CODE }
PERSON-NAME	::= { surname forenames	[APPLICATION 38]SET [0] OCTET STRING, [1] SET OF OCTET STRING
	OPTIONAL, title	[2] OCTET STRING OPTIONAL }
ADDRESS	::= { addr-lines	[APPLICATION 39]SET [0] SET {TERM-REF, OCTET STRING } OPTIONAL,
	postcode valid-from	[1] OCTET STRING OPTIONAL, [2] DATE }

HCF	::= { hcf-id name address contact-nrs netaddresses	[APPLICATION 40]SET [0] HCF-ID, [1] OCTET STRING, [2] ADDRESS OPTIONAL, [3] SET { TERM-REF, OCTET STRING } OPTIONAL, [4] SET { TERM-REF, URI } OPTIONAL }
UNIT	::= { term is-standard next-larger next-smaller system	[12] SET [0] TERM-REF, [1] BOOLEAN, [2] UNIT OPTIONAL, [3] UNIT OPTIONAL, [4] UNIT-SYSTEM OPTIONAL }
UNIT-SYSTEM	::= { name	[APPLICATION 41]SET [0] TERM-REF }
BASE-UNIT	::= { UNIT, unit-group	[APPLICATION 42]SET [0] BASE-UNIT-GROUP OPTIONAL }
DERIVED-UNIT	::= { UNIT, unit-group	[APPLICATION 43]SET [0] DERIVED-UNIT-GROUP OPTIONAL }
UNIT-GROUP	::= { name	[12] SET [0] TERM-REF }
BASE-UNIT-GROUP	::= { UNIT-GROUP }	[APPLICATION 44]SET
DERIVED-UNIT-GROUP	::= { UNIT-GROUP, derivation GER } }	[APPLICATION 45]SET [0] SET { BASE-UNIT-GROUP, INTE-
PHYS-REF	::= { storage-type storage-loc reference	[APPLICATION 46]SET [0] PLAIN-TEXT, [1] OCTET STRING OPTIONAL, [2] OCTET STRING OPTIONAL }
MOMENT	::= {	[13] SET }

```
DATE-TIME ::= [APPLICATION 47]SET
            { date [0] DATE OPTIONAL,
              time [1] TIME OPTIONAL }

CODE-LINK ::= [APPLICATION 48]SET
            { concept-code [0] OCTET STRING OPTIONAL,
              code-used [1] OCTET STRING OPTIONAL }

END
```


8 The GEHR Term Set

8.1 Introduction

The GEHR architecture and object model define the *Health Record Item* as the fundamental basic unit of health information within the EHCR, although complex constructs may require these to be organised within *HRI Collections* or under *Headings*. Careful consideration has also been given to the term sets which the architecture needs to accommodate in order comprehensively to address the needs of systems developers and clinical users.

8.1.1 Initial Clinical Evaluations

The early clinical evaluations of the GEHR project were based on a pre-existing prototype healthcare record application (Health.one, Health Data Management Partners, Brussels), which reflected some of the features appropriate to a healthcare record architecture. The application incorporated two multi-lingual term sets, one for *HRI names* and a second for the *HRI content values*. These term sets were subjected to an evaluation which took place amongst a sample of the existing users. An analysis was performed of the vocabulary used in a sample of clinical records on paper and computer in different institutions across Europe. The vocabulary used within the term sets of the major clinical classification systems (ICD, ICPC, and Read) was also studied.

This work revealed that the clinical classifications offered terms to a fine level of detail for HRI content values, although they did not provide a comprehensive range of HRI names. For example, ICD contained a very wide range of possible diagnoses, but did not contain the term *diagnosis*. Other examples of terms not offered by the classifications studied are *family history*, *subjective symptoms*, *on examination*, *management plan*, or *referral*. Many of these terms are used in healthcare records to organise the individual entries, or to provide the contextual relationship of a clinical concept to the patient (e.g. to convey if wheeze is a *symptom* or an *examination finding*).

These clinical classification systems also varied in the extent to which they provided terms for naming the general clinical concepts which might be included in a healthcare record entry. For example they would be likely to contain terms such as *essential hypertension*, *normotensive*, *sinus arrhythmia* and *consolidation*, rather than *blood pressure*, *pulse* or *chest auscultation*.

On a computer screen HRI names may correspond to the field labels on a form, and on paper may be explicitly stated as underlined or marginalised titles. However, in many cases these headings are implicit (or at least felt to be understood by any clinical reader) and clinicians are not always used to stating them. For example, a loose paragraphing arrangement on paper may be the only written structure to a basic new-patient examination, even though the information has been recorded in distinct unlabelled sections (such as family history, past medical and surgical history, allergies, social and occupational history).

The consistent use of such headings, or a clear cross-referencing between them, is vital if health information is to be shared across systems in the future. For instance, it must be clear if *penicillin* relates to an allergy for a patient, and not to a recent prescription actually given.

The early semantics work of the project therefore identified two aspects to the use of terms: some which would be useful as *HRI Names*, and those used primarily as *HRI Content values*. This latter group would include clinical findings, such as: numeric data; the terms currently found in most contemporary classification systems, such as diagnoses, treatments, clinical observations, symptoms provided by patients; and more complex data types such as images and drawings. Although some overlap in usage would occur, it was agreed that the reviewed classifications contained insufficient terms which could be used for *HRI Names*.

In order to support the early development of prototypes and to illustrate the features of the architecture with clinical examples, a term set was developed by the project itself. This work was not intended to replace the contribution of the existing classification systems, but to provide a term set which could be used by the prototype system developers to illustrate some of the features of the architecture.

8.1.2 The Final GEHR Term Set

The final term set, comprising 2,500 terms translated into 9 European languages, is included with this document as an ASCII text file. It provides a useful list of *HRI names* for which data about a specific instance (value) for a patient may be documented. In specialist areas GEHR will support the adoption of other term sets from emerging standards such as DICOM for images; some of these terms might be used as *HRI names*.

Compiling this term list presented many challenges to the semantics group of the project. It was considered important for the GEHR Term Set to be multi-lingual, in order to support the demonstration of the portability of the GEHR architecture across languages. A number of issues were debated within the semantics working group when assessing the inclusion of terms: plurality; gender; laterality with regard to anatomical sites; the qualification of terms with negative, probability, and severity statements.

Efforts were made to keep the term set as elemental as possible, but where complex constructs gave rise to a unique concept these were retained. Working in multiple languages gave rise to many instances where a composite term could be broken down into elemental constructs in some languages but not in others. A fuller discussion of these semantic issues surrounding this term set, and the strategies adopted by semantics group, is provided below.

The limited resources available for its development preclude its being regarded as complete, and at present there are no plans for it to be maintained as concepts and medicine evolve. It has been placed in the public domain as an example term set, and may be used by systems developers as part of their prototyping endeavours when using the GEHR architecture.

8.2 Semantic Issues

8.2.1 The Decomposition of Terms

A major issue for the semantics group was the extent to which the term set should contain long or complex constructs, as opposed to their elemental constituent terms. Examples of the former would be *date of first pregnancy*, *date of next childhood vaccination*, *age at onset of menstruation*, *chest auscultation on inspiration*, *systolic blood pressure on standing*. It has been widely recognised, for example from developing the UK Read term set, that there are an almost infinite number of such composite terms, and that no term set can or should seek to include many of these.

However, the early work of the semantics group revealed that there also had to be a limit to the decomposition of long terms into their constituent words or phrases. One possible criterion for this limit appeared at first to be the translation process. “Blood Pressure” cannot be decomposed into “Blood” and “Pressure”, because the assembled individual words when translated into French bear no resemblance to “Tension Arteriale”. “Blood Pressure” is therefore the minimal term which can be safely translated and maintain its clinical meaning. “Systolic” and “Diastolic” can be separated across languages from “Blood Pressure”, allowing the user to compose these specific terms as required.

A second point which arose was the problem of grammatically correct but clinically nonsensical combinations, such as “Systolic Blood Group”. It was agreed that the work of GALEN on clinical concepts and terminology was the more appropriate body to research this area, and that resources did not permit GEHR to embark on this vast task. The GEHR object model supports the combination of terms within the class for HRI names, but does not seek to enforce clinical sense.

One proposal was for terms not to be broken down beyond a particular level in each language (e.g. *maiden name* cannot be broken down into *name of maiden* in English). The fragmentation of terms could stop when the concept cannot be further broken down in any one of the languages included in the project. This would assure a one-to-one mapping relationship throughout the translation process, but had the drawback that for any new language all of the concepts would have to be considered again. An alternative was to break each term down as far as possible to the minimal concept level independently for each language. A mapping framework could then assist the translation in situations where concepts do not exist similarly in all languages.

For example:

type of housing	<=>	type d'habitation
-----------------	-----	-------------------

could become:

housing	<=>	habitation
type		type

but that in French:

habitat	=	habitation
		type

French clinicians may usually prefer to use the word *habitat*. However when *type of housing* is translated from English, *type d'habitation* could be the “preferred translation”, thereby minimising any risk of errors, and maintaining the original Collection structure (and the relevant attributes). *Habitat* is treated as a synonym in this case.

This would allow:

housing	
type	
size	
ownership	

to be translated into French without any loss of structure. If *habitat* and *type of housing* had been the minimal bilingual concept, the above Collection could not have been supported across the two languages.

When considering the decomposition of terms, and their subsequent recombination in different ways, some compromises had to be made. The GEHR Term Set needed to contain sufficient elementary terms in order to allow flexibility for users to compose their own specific terms. At the same time, in order to make the term set useable in the test sites, frequently-used complex terms or those which have a unique conceptual meaning were also retained.

This will inevitably mean that some phrases exist in full in the term set, but could also be composed from element terms; e.g.

Date of Birth	<->	Birth	
			Date

There are two alternative mechanisms for coping with this.

- The knowledge exists at a software (data entry) level, forcing the data to be stored using the most elemental terms (in the above example, it would mean a user entering data as shown on the left would have it changed by the system to store it within GEHR as shown on the right).
- GEHR-compliant applications store any or all of these constructions faithfully, and that it is up to the viewing and query tools of software systems to recognise that they are equivalent representations of the data.

The semantics meeting favoured the latter approach, because:

1. data recording is then faithful to the users original view;
2. GEHR is not required to produce a preferred term construct for every single clinical concept, which is a vast task for which other projects have been better-resourced.

The consequences of such a decision, however, is that the codes for:

Systolic Blood Pressure	120
Diastolic Blood Pressure	80
Blood Pressure	
Systolic	120
Diastolic	80
Blood Pressure	120/80

would bear no resemblance to each other, and that GEHR at an architectural level would be unable to recognise that these were different representations of the same data. It would be the application that would study records containing a mixture of such representations and be able to present the user with a consistent visual display or analysis.

This approach also simplifies the translation process. Each term can support a specific translation into other languages, but where a complex term has been composed from elementary ones it would not be a requirement of GEHR to guarantee a perfect translation of the construct: each term element could be translated separately, and an application query tool would need to recognise the meanings attached to the composite. This is a major piece of work, more properly in the domain of GALEN.

A parallel case may be presented for acronyms such as *Colle's Fracture*, which could also be seen as a synonym for the fuller definition (which would describe this special type of fracture of the wrist). This again protects the translation process, in which acronyms should be mapped to their full description for the translation of the component terms.

8.2.2 Synonyms

Synonyms become very complex when evaluated across languages. It is clear that in the majority of cases (as documented in Deliverable 4: the Requirements for Clinical Comprehensiveness) abbreviations probably represent the majority of such synonyms. Others are likely to differ from a preferred term by only a subtle difference in meaning. It is clear that in order to suit most users some mechanism for handling synonyms will need to be offered. This is largely the domain of the term set developers. It is possible that in some situations synonym lists will be used by applications to help users navigate to a preferred term.

8.2.3 Plurality

In many instances the singular or plural use of a term conveys significant meaning, e.g. *pain in the hands* implies it is bilateral. GEHR must be able to record if a singular or plural term has been used, and in order to offer a readable form of the record the GEHR Term Set will need to hold both singular and plural forms of each term (e.g. *phalanx and phalanges*). The GEHR architecture incorporates a mechanism by which singular and plural terms can be distinguished.

8.2.4 Gender and Plurals

Gender is rarely an issue in the UK, and in other European countries it is uncommon for the same word to exist with completely different meanings differentiated solely by the gender used. The main exception to this is when people are being described. *Nurse* is a good example to consider:

Nurse, Nurses
Infirmier, Infirmiers, Infirmière, Infirmières

In most cases, the user will not wish to be emphatic about gender or plurals when recording the actions of a nurse. Systems should not force the user to decide in cases where the clinician might not wish to record this and where the language is ambiguous (such as English). In French, if a system offers the user a choice, the recording of this might be:

Root term: Infirmier(e)(s) Gender: Female Plurality: Singular

This would translate directly into English, but the system may not bother to show the gender and plurality unless the user requested this information.

Users and their systems may wish to support the specific words *Infirmier*, *Infirmiers*, *Infirmière*, *Infirmières*, in which case these might be classed as synonyms used in that particular recording instance. This might be the same, for example, in the case of *phalanx*, *phalanges*, where a term set or system developer might feel that the use of the root term alone in a plural case would look clumsy. They may then wish to incorporate a more sophisticated set of such synonyms.

8.2.5 Laterality

For the majority of anatomical sites it is preferable to avoid specific terms for the left or right part of the body. Such a specification would unnecessarily add to the size of the term set and force the user to specify a side in every instance. Plurals is one case where expressing laterality is redundant (*swelling of the ankles*).

However, many of the larger organs have different characteristics on the two sides: the left lung has two lobes whereas the right one has three. There may be a valid case for supporting the specific terms *left lung*, *right lung* in order to allow system developers to incorporate validation measures and, in the future, appropriately integrate decision support tools.

There are (at least) three fundamentally different ways in which, for example, 'left' could be used:

1. As a qualifier of an anatomical (mostly) or physiological thing

e.g. left shoulder
 left bronchus
 left hemisphere
 left head (2-headed Siamese twins only)

Here the purpose would be to make a choice between two available anatomical objects of which two exist in the body, a left one and a right one.

Grammatically, left and right would be adjectives qualifying the anatomical name (noun). This probably applies across languages: *la main gauche* etc.

2. As a way of denoting the location of something.

e.g. the pain is on the left

This is broadly equivalent to:

 the pain is left sided

but it may be necessary sometimes to distinguish between on the left and throughout the whole of the left. Grammatically this is a noun form: *on the left*. In French this would be *sur la gauche*.

3. As a way of denoting the direction of something

e.g. a. pain radiating to the left
 b. a knife wound starting at the navel and continuing for 5cm to the left.

Grammatically this corresponds to an adverb modifying a verb (usually a verb of motion). In French *à gauche*.

The term set therefore contains 3 terms for left and 3 for right etc. All necessary combinations can be formed using other terms which occur only once in the term set *hand, bronchus, hemisphere* etc.

8.2.6 Other Anatomical Sites

The use of anatomical qualifiers such as *upper, lower, distal, proximal, medial, lateral* also need to be considered when drafting the term set. In some instances, such composite terms will have a unique meaning at a linguistic or concept level: the *upper leg* is the *thigh*, expressed as one word in both English and French. However in most cases the term is used to describe a part of an organ rather than a new more specific organ.

The debate about specific composite terms resembles that for laterality.

Notions of up and down:

choice	upper, lower
location	upper, lower
direction	upwards, downwards

Notions of nearness:

choice	proximal, distal
also	near, far
location	proximal, distal
direction	proximally, distally

Notions of single- double-sided:

choice	-
location	lateral, unilateral, bilateral
direction	laterally, unilaterally, bilaterally

It should also be remembered that words like *thigh* are synonyms, in this case for *upper leg*. It was difficult to resolve to what extent the GEHR Term Set should support the former, the latter or both.

8.2.7 Other Qualifying Terms

It is clear that some rules must be developed to cope with terms which further qualify their root term. Examples of this include *first*, *last*, *latest*, *next*, and might be combined with *date of* to give terms such as *the date of the last smear*.

Similarly, *highest*, *lowest*, *heaviest*, *lightest* are concepts which may be used by systems for analysis queries (such as the result of the *highest systolic blood pressure* in the record).

8.2.8 Conclusions

The decisions taken by the Semantics Group on the decomposition of terms have placed some interpretative responsibility on the software application, or any future concept classification system. GEHR is not seeking to enforce conformity through a restricted term set or combinatorial rules. However, we hope that by facilitating the sharing of health records clinicians will learn from reading each others notes and that a consensus of good recording practice will eventually emerge.

8.3 Evolution of Term Sets

The GEHR architecture has been designed from the outset to cope with the recording of terms from any conceivable term set, whether coded or not, whether registered with an agency, or local. However, it is important to be aware of the limitations of the use of term sets in general.

Although the GEHR approach permits the display of any term where either its code is recognised

(i.e. from a code set known to the system in which at least that code is present), or its textual expansion has been explicitly recorded, there is still no guarantee that such a term can participate in useful querying or analysis on any particular system.

Consider that the term 'Myocardial Infarction' is recorded, perhaps as a code from a recognised code set. If this is received by a site at which that code set is not handled by the system, it would not be possible to use it in a query such as 'show me everywhere that a heart attack is recorded'. In fact, if the code set is not known, it may not even be possible to ask for 'show me all Myocardial Infarctions' since this term will have different codes in different coding schemes. If the term has been expanded, it might be possible to compare the textual form but there is no guarantee that such a search would produce the correct results: variations in spelling or, in the wider European context, recording in a different language may make the task impossible.

One way of using terms in queries is either to adhere to terms from a single code set or to cross map all those known to the system. Neither can be guaranteed in the context of portable, communicable records and the latter option is liable to be an ever increasing task.

The only safe way to guarantee that terms are not just recognised (for display) but interpreted in a consistent way across all systems, is to have a single, standardised set of medical concepts to which all term sets/coding schemes can map, each with a clear and precise medical meaning.

To make full use of this in the European context, each of these concepts will need to be expressed as terms in each of the European languages.

Extensive work has already been done on term sets and coding schemes (e.g. ICD-9/10, ICPC, UK Clinical Terms) and the GEHR project itself produced an example term set, translated into nine languages, to demonstrate the concepts. However, the investigation and research required to produce a Europe-wide, standardised set of medical concepts is well outside the scope of the GEHR project. Nonetheless, collaboration between projects like GEHR and the projects involved in research and development of medical coding schemes will be invaluable in the future in facilitating the production of such a coding scheme.

It should also be recognised that a similar problem exists with the interpretation of Units. The GEHR architecture provides for Units to be recorded as terms from any term set but, again, in order to process units properly, including unit conversions, it is necessary to know the nature of the Unit concerned.

Whereas, for terms in general, there may be many thousands of concepts which need to be expressed, terms for units are far more limited. There is, for example, complete agreement on what a kilometre or an hour is and how these relate to miles or days respectively. There are well known rules for extending units, for converting units and for abbreviating units.

A complete units model could be constructed, implemented and made available to all users of the GEHR architecture and it is hoped that this may be accomplished in future work. In the meantime, the minimum information required for an implementation to understand a unit has been modelled.

Many observations that might be recorded will have a limited number of normal qualifiers (e.g. hand will normally only be left or right), a natural type of content (e.g. surname would normally be recorded as a piece of text, weight would normally be recorded as a quantity of some kind

perhaps with additional terms or text), and possibly a natural set of units (Unit_Group) (e.g. weight, if recorded as a quantity, should be recorded in units of mass).

Such guidelines and extensions to the way in which observations are recorded is beyond the scope of the GEHR project. However, future projects should seek to address these issues in a manner consistent with the GEHR Architecture.

9 Implementation and Tools

9.1 Implementation Aspects

The purpose of this Chapter is to provide guidelines on how to translate the semantics captured in the model to the various implementations found in real-world systems.

9.1.1 Libraries and Applications

As stated in earlier Chapters, it is essential that any model be implementable, and that the path between a model and implementations be as seamless as possible. A number of points apply to all types of implementation:

- it is crucial that the mapping between the OM and the implementation be as automatic as possible. Conversions involving significant manual checking and hand-coding are prone to errors, and can only be validated by eye;
- it is preferable to adopt an integrated engineering approach with some OO content, rather than slavishly try to map the OM to lower level languages for all parts of an application.

At sites where non-OO software is generally written, and non-OO data storage is used, there is often the impression that one cannot really take advantage of OO techniques. The supposition that object-orientation must supplant all other information technologies is far from the truth. The most basic approach is to realise for which parts of applications freedom does not exist to re-engineer into OO. This may include the database interface, GUI, network interface, etc.

9.1.1.1 Object-Oriented Library Approach

The most desirable approach is to build an OO library based on the GOM, which contains the processing logic for all objects in the model.

9.1.1.2 Integrated Applications

The core of an application can be rewritten in an OO language such as Eiffel, Ada 9x, C++, Objective-C and possibly Smalltalk. This can then be compiled to a different programming language (e.g. C), or could be compiled into a library which can be interfaced to another programming language. Most GUIs such as Visual Basic, and many database languages such as Oracle, INGRES, Informix, Sybase, Microsoft Access, etc. have published interfaces to other programming languages.

The advantage of having an OO core to the application is that the main logic corresponds very closely with the Object Model from which it is derived. Thus applications can be engineered with a thin veneer of logic calling into the OO part, which itself makes calls to both the database, to access information, and the GUI, to display results. Control can be in the hands of either the OO or the GUI part (one of them normally contains an event loop).

The problems of rationalising the OM with respect to non-OO databases do not go away, but they are now directly in the hands of the developer, and can be solved largely in the OO part of applications.

An advantage of this approach is that if and when GUI or database is upgraded, replaced, etc., the core of the application is largely unaffected.

9.1.1.3 Incremental Re-engineering

Another well-known approach is to rewrite completely one application at a time (usually starting with the least risky one) and gradually replace any or all applications for which the OM is relevant. Many OO development environments will provide both complete GUI and relational database interfacing (enabling an extract of a database record to be read into an object instance in memory), and can build complete applications. There are environments in Eiffel, Ada 9x, C++, Objective-C, Smalltalk, and many others which will do this. The advantage of this approach is that OO development and learning is carried out incrementally, rather than in a revolutionary way.

[It should be remembered that not all applications may need knowledge of the GOM; typically RDBMS administration programs are perfectly functional as they are.]

For applications to be written largely in an OO language such as Eiffel, Ada 9x, SmallTalk or C++, the mapping from the GOM is likely to be reasonably simple.

9.1.2 Data Storage Implementations

There are several interesting products relevant to ODBMS available for PCs, although further research would be needed to determine which might be best for use with the GOM. There may be technical problems involved in conversion from the GOM to a database schema language, although since the conversion is basically OO to OO, these should not be insurmountable.

For relational databases, the approach is to:

- set up the relational side to reflect closely the class structures in the object model.
- develop rules which define the transformations from the OM to relational tables and build an automatic parser to generate the entire schema from the GOM.
- use the same approach for existing databases and green field sites.

A lot of work is required to prove this approach, but OO concepts including multiple inheritance have been successfully implemented in a relational environment. This plus the modelling of aggregation from the OM could be done through the use of view definitions, whose attributes are finally taken from existing base tables, or from new base tables where they did not previously exist.

Mapping is required between the base types of the application language and those of the database. Normally, this should be quite easy. One approach is to start with the OO Programming Language types and simply create standard definitions for how they are stored.

Base View: may actually be a base table, or simply a view which collates relevant attributes from existing tables in the database. Provides a place where class & attribute name changes can be made with respect to existing names in DB;

Association View:

maps associated tables together; aggregation differentiated by cascade semantics rather than break-link semantics;

Inheritance View:

maps rows in tables corresponding to descendent classes to the appropriate rows in parent class table(s);

There is a basic mapping from class structures to tables, utilising a type attribute introduced into all relational tables.

9.1.2.1 Association, Aggregation and Multiplicity

Associations between classes in an object model correspond to any relationship between them that is not inheritance. The association relationship also corresponds directly to the idea of a relation expressed as a tuple (and implemented by a join) in a relational database, so there is no problem adding this to the relational model.

Associations exist between instances at runtime, and have a multiplicity which can be described in the object model. Multiplicity is the characteristic of an association defining the number of instances allowed to take part on either side, and is commonly expressed as **1:1**, **1:n**, or **n:n**. The concept of the multiplicity in a relational model is virtually the same.

Aggregation in the object model corresponds to the *part-of* relationship between a parent object and sub-part object(s), and is a kind of association that occurs so often it is useful to treat it explicitly. Typical examples of aggregation are the parts-explosion trees corresponding to real-world objects such as cars and aeroplanes. The distinguishing characteristic of aggregation as opposed to other kinds of associations is that functions and operations executed at the parent propagate to the sub-parts.

In going from the object model to the relational, classes must be dealt with separately from the relationships between classes. An association between classes in the object model corresponds to links between instances at run time. In relational terms, these links will be foreign keys. The location of the foreign keys depends upon the multiplicity of the relationship, as follows:

- 1:1** either or both of the related entities; usually choose the one which is the subject of querying;
- 1:n** in the entity on the “n” side, or in a separate relation table to facilitate querying;
- n:n** in a separate relation table;

The semantics of these primary/foreign key links must be defined further, in order to properly model the association and aggregation concepts. This is specified in terms of the *link type* of the relation, as follows:

- n** association is modelled by *break-link* semantics (i.e. break foreign-key links on deletion) or *disable-link* semantics (i.e. do nothing if associations exist and flag deletion attempt to user or administrator);

- n aggregation corresponds to *cascade-link* semantics (i.e. follow foreign keys recursively on some operations, including creation, deletion).

The mechanism by which the link is maintained must also be indicated. There are two possibilities: standard primary/foreign key relationships, indicated by the <- sign, and assignment, indicated by the <= sign. These do not change the relation semantics in the object sense, but define how the values at either end of the link are to be maintained.

9.1.2.2 Inheritance

Single Inheritance: Inheritance in an object model corresponds to the *is-a-kind-of* relationship between classes. There are a number of ways this can be done in a relational database:

1. use a table for the superclass containing just those attributes defined in the superclass, and to create separate tables for each subclass, whose attributes are those unique to that subclass. To facilitate joining, two things can be done:
 - a) foreign keys for the superclass table rows must be placed in the subclass table rows, or;
 - b) the superclass table must contain a foreign key column containing the key value of each subclass table record.

Which method is adopted depends on whether accessing is being done at the superclass or subclass level.

This method is intuitively close to the OO model, and should lead to easier database maintenance.

2. define a table for the superclass, whose attributes will include all those of all subclasses, plus another attribute to differentiate on class. This results in a table where some space is wasted.

Both methods are useful; the decision of which one to use depends on how alike the subclasses and the superclass are: if the subclasses only add a few attributes to a large number already defined by the superclass, the all-in-one table may make more sense. Alternatively, if the subclasses are quite different, and the superclass supplies only a few attributes, the multiple table mechanism is better.

Multiple Inheritance: requires a set of rules to be established whereby the full relation definitions can be generated automatically from the model. Such rules would address:

- link type;
- indexing (determined by querying the requirements of applications);
- functionally derived fields;
- correct handling of multiplicity.

This Section has necessarily been brief. Nonetheless, it has served to illustrate the practicalities of implementing a system based upon the GOM and the kinds of problems likely to be encountered. The bibliography includes a number of useful books which may provide further information on some of these topics. In addition, it is hoped to include details on an exchange implementation in

future documents.

9.1.3 Disclaimer

Every effort has been made to check the correctness and consistency of the GOM described in this document. However, until it is produced in a validateable form (a subject of Future Work) it is impossible to guarantee correctness (except for consistency within itself). Likewise, every effort has been made to ensure consistency between class diagrams and the formal text - if any disagreement is found, the formal text should be taken to be correct.

It should be noted that the Eiffel short form to mark-up language translator is not perfect, and occasionally generates the wrong font instructions etc. Although every effort has been made to correct these errors, it is possible that some may have been missed. Only the appearance of the text is affected.

This is a long and complex document and although every effort has been made to ensure consistency and correctness, there may be errors.

Specific queries should be addressed to the GEHR Office whose address appears at the front of the document.

9.2 Tools

Deliverable 19 is concerned principally with the documentation of the GOM and with an explanation of the evolution of the GEHR architecture and the GEF. Certainly a great deal of effort was focused upon the architecture and systems areas during the GEHR project. However, collateral with these efforts, considerable work was being devoted to the development of a number of software tools and utilities.

This tool development work was intimately tied up with, indeed dependent upon, the developing architecture. The initial objectives of the tool work were to tackle data query, integration and conversion, that is, the communication of data to and from GEHR. In order to achieve these objectives, a practical approach was adopted within the consortium, a specific methodology based on the evolution and testing of prototypes within a comprehensive framework of requirements.

The experience gained in the early phases of the project from experimentation with an existing architecture, the Current Health Record Architecture (CHRA), together with extensive research carried out into user requirements and requirements in other, associated domains contributed much to the synthesis of the functional requirements for GEHR. The principal requirements (for clinical comprehensiveness, for portability and communicability, etc.) are summarised in an earlier chapter in this Deliverable. These were a primary influence upon all development work including tool development.

The development of various GEHR tools and prototypes has been reported in a number of Deliverables, two in particular: Deliverable 11 'GEHR Interim Software Tools - Interim Specification' [26], and Deliverable 13 'Final Systems Report' [27]. The former Deliverable

reports the early experimentation with tools, the latter covers the more advanced stages in tool evolution and the gradual emergence of prototype GEHR compliant systems.

The initial experimentation with tools was based on the CHRA. Interim tools were developed to answer the specific needs of data communication with the CHRA. This experience was instructive, leading to the development of principles which could be applied as the evolving GEHR structure progressively replaced the CHRA within the project, resulting in a more elaborate toolset. The original intention was to work first on the lowest level access and integration routines, then to explore further tools, encompassing a wider context.

A key feature of the GEHR project has been the institution of a network of clinical test sites throughout Europe to be used for the evaluation of successive generations of prototypes. The importance of these sites has been emphasised from the outset; they have engendered a lot of interest from other projects (including other AIM projects). Their purpose is to allow pragmatic assessment to be made of the draft architectures and supporting software tools. Furthermore, such assessment is undertaken by practising clinicians, the target users of the eventual systems and therefore the best judges of user satisfaction.

The evolution of the architecture, the gradual move from the Interim GEHR Architecture (IGA) to the architecture expressed in the formal GOM has certainly affected tool work. The development of this trend, the undoubted potential offered by the Object Oriented approach and the advantages associated with it, are thoroughly documented in this Deliverable.

The tool development work was simultaneously dependent upon this evolution and influential upon it. Experimental work undertaken by tool developers was used to investigate particular areas or topics, feedback from the results of which was used to refine the structure of the evolving architecture. Thus, although the GOM did not fully evolve until the final year of the project, the experimentation yielded valuable principles which can now be applied to the GOM and GEF.

The nature and format of the toolset has itself been influenced quite markedly. There is an obvious distinction between the toolset as originally envisaged (and as portrayed in the initial workplan), and that which has finally developed. Much of the basic functionality is now implicit within the current GEHR Object Model which presents an optimal way of implementing this functionality.

Deliverable 16/17, 'GEHR Software Tools and Prototypes / Application Software Integration' [28], report recent demonstrations of GEHR specific tools and prototype systems which have been designed and built to exercise the architectural principles. The aim of these demonstration was to show, by using the tools and systems to manipulate particular clinical examples, how certain key features of the GEHR requirements have been tackled. An additional intention was to display effectively the close integration of the various systems and tools and the ease of communication between them.

These Deliverables demonstrate the considerable effort that has been dedicated to the assessment of the developing architecture. The experimentation with tools and the attempts at integration have contributed substantially to the successful evolution of the architecture.

The GEHR Brochure summarises what has been achieved. The project has built and provides specifications for tools to interface and integrate the GEHR architectural formalism (exchange format and object model) with the emerging range of standard messages and architectures from

specialised domains and systems. Examples include laboratory data (OPENLABS), images (DICOM), ECG (SCPECG), prescribing (OPADE), systems (HELIOS) and such generic messages as HL7 and communications standards such as EDI.

Details of the current specifications and state of development of the various tools are presented in Deliverable 18 [12] but, for completeness, a brief summary of the work is included below.

9.2.1 The Clinical Drawing Tool.

One particular product which has resulted from the tool development work is a Clinical Drawing Tool. This has been developed by the team at St. Bartholomew's Hospital Medical College and utilises a comprehensive library of anatomical drawings which have been designed by GEHR and placed in the public domain.

Deliverable 4 [5] specified that, in the move from paper-based to Electronic Health Care Records (EHCRs), clinicians must not lose the ability to record certain types of data through the use of drawings. This implied:

- The availability of a tool (outside the architecture) to cover aspects of input, review, modification of the data and to make the data available for incorporation into the EHCR.
- Suitable features in the GEHR architecture to accommodate such data.

A set of standard Library Pictures has been included with this Deliverable on an accompanying floppy disk. A demonstration version of the drawing tool is also included on the disk, enabling the individual pictures in the library to be viewed. The copyright for the full tool is held by the GEHR Coordinating Partner.

The library provides a number of benefits in the recording of data and the transferring of data. It:

- makes it easier to create drawings since a user will not have to draw the underlying, anatomical library picture upon which Drawings are made;
- promotes better quality communication of Health Record data as all drawings will have a common basis;
- makes it possible to transfer data between different computer platforms as the library pictures and markings will be independent of operating system.

The Drawing tool will aid clinical diagnosis in two ways:

- The use of overlays of medical sub-systems (e.g. dermatomes, skeleton) correctly aligned on a library picture/drawing will help the user in making inferences from the drawing. These can then be recorded using the drawing tool.
- By using medical protocols (e.g. Ritchie score for joints), programmatic deductions of abnormalities can be presented to the user, and can then be recorded.

The drawing tool may be linked to a clinical database application or may be a stand-alone application. It allows users to create a drawing or drawings from their observations of a patient and to view or modify a drawing or drawings previously created.

To create a drawing, users retrieve a library picture and then add markings to the picture. Markings can be added to express the *type*, *extent*, *location* and *severity* of the observations. Free text and/or coded terms may also be used to form a label for markings; this annotation will always be shown on-screen.

Viewing an existing drawing or drawings results in a display of the drawing or drawings on-screen with the library picture, markings and summary correctly displayed. Users can manipulate each Drawing and read the label associated with all markings made.

In accordance with the various requirements and the GEHR model, a drawing cannot be saved without storing the identity of the patient about whom the observations were made, the user who created the drawing and the date, time and location that the drawing was created. These data must be present or accessible when viewing and may be present and recorded in the EHCR (if not, the drawings application must record them).

9.2.2 Image Tool

The same team, in conjunction with specialists from SmithKline Beecham, devoted a great deal of research time and effort to the examination of ways of handling and referencing clinical images and biosignals within the EHCR. The EHCR must be able to contain, or to reference, image results (mainly X-rays) and these must also be transferable between clinical sites.

This work has determined the main goals for an image tool:

- to incorporate images into the EHCR;
- to improve image handling by clinicians (i.e. to allow clinicians to view images and not just reports based upon them);
- to allow access to images stored elsewhere (because of storage problems due to size of image data).

The work has considered the fundamental concepts involved in drawing up a specification for an appropriate handling tool. These have included the way an image tool should interact with a local EHCR and a mailbox system, and how it should mediate the request result cycle. Storage implications have been tackled together with mechanisms for achieving compression of data. The methods of viewing an image have been explored, the probable use of overlays and the requirements for the import and export of data.

Relevant standards and protocols were investigated, the need to handle data in common formats for medical images, such as DICOM, JPEG, IIP-IIF, etc., also the requirement to link to EurIPACS systems.

A number of recommendations as to an appropriate specification for an image handling tool have been made, based upon the results of these comprehensive investigations. These are presented in detail in Deliverable 18 [12].

9.2.3 Mailbox Server Mechanism

The transmission of sensitive, clinical data in the form of an exchange format requires a secure transmission medium and method. Messaging using mailboxes as a suitable method of communicating health data within the scope of GEHR is considered at length in Deliverable 6, GEHR Requirements for Communication.

France Telecom (FT) has been developing its work in the area of added value services and has proven expertise in this area. A number of communication products and services for messaging are available commercially. Although these are of general application, FT's mailbox server mechanism and programming interface (Messaging Services Interface) were seen as offering great potential for GEHR, especially to support experimentation in communication between various elements in the emerging network of test sites.

The MSI provides heterogeneous messaging and an integrated, complete solution for the end user. The system has a number of important advantages:

- messaging is built upon standard X.400;
- it allows for the possibility of EDI;
- asynchronous communication allows each user to work in his/her own mode;
- ready availability of products.

The MSI provides direct and simple access to mail services, enabling clients to communicate with each other in store-and-forward manner. The underlying mail system takes care of the routing and delivery of message.

The mailbox system was available (potentially) to all consortium members together with comprehensive documentation to inform users how the system could be integrated into their test site operations. Although there was insufficient opportunity during the project to take full advantage of the potential offered, it is anticipated that extensive use will be made of this facility in any follow on work.

9.2.4 GEHR Upgrade Tool

C2V spent a great deal of time examining the requirements for portability (i.e. to work on a variety of platforms), and the consequent need for data representation to be independent of any particular data storage and access method. This led to the development of memory caching mechanisms and a virtual memory manager.

The preparatory work culminated in the development of a File Container software library, designed to handle data storage and manipulation on a file system and provide easy communication with existing relational database management systems. These facilities have been successfully demonstrated upon a number of platforms, from PCs running DOS and Windows, to SUN workstations running UNIX / X-Windows.

The later work focused on the relationship between the GEHR architecture and its implementation. The File Container software library was extended, the principal concerns being

the storage of data, the arrangements for data transfer, and how to 'track' the organisation (structure) of data. The approach adopted has already been documented in Deliverable 13 [27], which reports how the initial design of the File Container was analysed and modified from the performance and security perspectives.

The later development, in preparation for the demonstrations at the end of the third year of the project, involved movement away from the initial testbed to a more general use for the File Container: a file support mechanism for storage and transportation of GEHR compliant data.

There was considerable collaboration with the University of Hull and Microdata to facilitate storage of data and operation at the GEHR exchange format level. The focus was on providing a GEHR upgrade tool to offer conversion of non GEHR-compliant medical data to GEHR Exchange Format data. One system selected for initial experimentation and assessment was Microdata's Medicort system; manipulation of data from this package constituted part of the final year demonstration of tool capability. Deliverables 16/17 [28] summarise the essential features of the demonstration. Deliverables 11 and 13 [26, 27] give background information on the package and the current status of the tool is reported in Deliverable 18 [12].

9.2.5 Communication with Other Systems and Standards

The most cursory of surveys of computerisation within clinical institutions will reveal that there is a great deal of money and time invested in existing legacy systems. Two of the project participants, Kalamazoo and Microdata, devoted considerable research effort to an examination of the ways GEHR could interface with such legacy systems, both in hospital and general practice situations. The objective was to facilitate upgrade from such systems and to provide secure and efficient data portability from legacy systems to GEHR systems. The two companies sought to establish generic principles in preference to specific solutions.

This work was not carried out in isolation. There was considerable collaboration and co-operation with other participants in the project, for example Kalamazoo with HDMP and Microdata with C2V. In addition, there has been consultation with other software companies.

The team at Kalamazoo selected a number of legacy systems to investigate and assess: nursing, administration and medical file systems (mainly hospital based). The object of the investigation was to determine how to facilitate real time communication between such diverse systems.

The conclusion was that several 'levels' of tool are necessary, firstly to buffer communication between systems, in effect, to produce for each system a "mail box in" and a "mail box out". Secondly, it is necessary to develop tools to handle / manage the correspondences between the different systems.

Essentially, the structure of the mail box depends on the structure of the application. It is acknowledged that this will need a certain amount of modification to each piece of software but it is hoped to minimise the extent of this. The intention is that changes should only be made once, the buffering meaning that developers do not have to write new interfaces for each system they wish to communicate with.

Microdata has been examining the communication of medical data, laboratory data and images

between clinics, general practitioners' surgeries and laboratories. As already indicated, the objective has been to develop generic principles for the upgrade of data from legacy systems to GEHR compliance. In pursuit of this objective, an attempt was made to define classical data structures. The conclusion, reflecting the wide variety of data structure handling systems, was that, in the context of the GEHR project, a classical data structure should be regarded simply as one that is not based upon the GEHR type of data model. In other words, a common standard is required for communication of data across applications.

The methodology adopted was to establish communication of data between the various test site partners according to the exchange protocols needed, thereby to establish principles which could be applied to the emerging, definitive GEHR protocols. The task was split into a number of phases:

One objective is to access and store images in real time. A software tool is being developed to allow the capture, freezing, storage and retrieval of images (TIFF format at first, later DICOM), these images to be accessible by Microdata applications first, then communicated via additional tools to evolving GEHR applications. (Also, albeit at a later stage, the images will be communicated between the various software applications).

Another objective is to have exchange of data amongst general practitioners, a radiology department, a gastroenterology department and a laboratory. The need for a common data exchange format is explained above. The Microdata team is intending to use EDI protocols to convey messages in such an exchange format. The software tools should be capable of data integration and conversion between the several existing systems (COBOL files from Microdata on DOS 6.0, files from AS400, Word 4.0 reports on Macintosh, DBase laboratory files), i.e. between Microdata applications and other data files.

Microdata has collaborated extensively with C2V in this area. Deliverable 11 [26] reports the early stages of this work, the capability offered by C2V's FileContainer software for porting COBOL ISAM data, together with investigations into its potential for free text indexation and as a browsing tool. The more recent collaboration has examined the FileContainer access method for ASCII, COBOL, MEDICORT and PARADOC files. This work has culminated in the development of C2V's prototype GEHR upgrade tool which is described above.

9.2.6 Conversion Tools

The work of Health Data Management Partners has been influenced by similar requirements. The team concentrated upon the development of a number of conversion and integration utilities to facilitate communication between GEHR and non-GEHR systems/messaging formats. In order to assess these utilities, the team established close working relationships with a number of key test sites.

A great deal of experimentation was done with interim exchange formats before the emergence of the definitive GEF. This experimentation was extremely valuable in proving the appropriateness of the design approach and the effectiveness of the algorithms selected. In the first phase, the assessment involved the communication between different standards - i.e. attempting to recognise the emerging industry and specialist standard protocols and certain (non-) standard data formats and the CHRA format. Useful principles were established which can now be applied to the

prototyping of conversion tools using the GEF as the exchange format.

As far as communication between external information providers and a medical record system is concerned, the following tools were prototyped:

- CHRA_GEF conversion of CHRA format into GEF;
- GEF_CHRA conversion of GEF into CHRA format (i.e. the Health.ONE intermediate exchange format allowing integration using a standard conversion module;
- EUCL_GEF conversion of the EUCLIDES format into GEF;
- ILAY_GEF conversion of the ILAY format to GEF;
- GEF_ILAY conversion of GEF into ILAY format;
- GEF_RMIF conversion of GEF into RMIF, i.e. an integration format allowing the integration of data into the MUMPS based HIS of the Royal Marsden Hospital.

[In addition, a number of straightforward utilities were written to assist the development process. These included simple converters such as to/from ASCII format and to printable format.]

Two different approaches were adopted in developing tools depending upon whether or not the architecture of the medical system was based upon GEHR principles. A particular objective was to try and define a generic approach to the translation or conversion of data from a non-GEHR format to GEHR compliant format (and vice versa).

The general principle adopted was to allow different systems to talk to one another and to exchange data without having to write a specific communication protocol for every two systems X(i) and X(j). In other words, instead of converting from X(i) to X(j) directly, the conversion should proceed via Y (i.e. X(i) to Y and then Y to X(j) and vice versa) - where X(i) is, for example, the format used by a clinical laboratory, X(j) is the format used by a medical record system and Y an intermediate format. The GEF is an ideal candidate for this intermediate or exchange format role. (This reinforces the conclusions reached by Kalamazoo, etc. above).

There has been concern throughout the GEHR project to try and achieve consistency with existing and emerging standards and protocols in order that GEHR systems can integrate easily with the other elements in the modern telemedicine context. For instance, HDMP has worked closely with participants in the AIM OPENLABS project to provide and foster easy communication between health care providers and clinical laboratories. The list above includes tools which have been developed to provide conversion between the GEHR Exchange Format and the standard ILAY format messages. The intention is to achieve integration of laboratory data from an ILAY message into a GEHR type medical record - consistent with the original GEHR work plan to provide data integration.

HDMP recognised the great need for rigorous assessment in a real clinical context and made substantial efforts to set up test sites offering communication links between laboratories, hospitals and GP clinics. As the various GEHR features became apparent, they were integrated into prototypes to be used and assessed in operations at these various test sites.

9.2.7 Summary

This brief summary gives some indication of the breadth and depth of tool development work undertaken. Further details of the current specifications and state of development of the various tools are presented in Deliverable 18 [12]. This experimentation has been extremely fruitful and feedback from the results of the investigations has helped to guide the evolution of the architecture.

Deliverable 18 demonstrates that several of tools are in the process of being themselves upgraded to reflect recent refinements in the architecture - the feedback is two way in nature! Where necessary, recommendations for changes are included, and the Deliverable highlights areas where further work is required, and/or where new features may be needed, etc. It must be remembered that there has not been opportunity within the timescale of the present project to subject these tools and prototype systems to intense scrutiny. Further changes will no doubt become apparent as the software is subjected to rigorous assessment at the test sites. In addition, clinical and other feedback from operational experience with prototype compliant systems in any follow up project(s) will no doubt influence tool specifications.

10 The Way Forward

10.1 The Future

The derivation of the GEHR Object Model and the definition of a suitable exchange format have been major achievements of the project. The GOM and GEF only emerged towards the end of the project, however. It must be remembered that much of the experimentation during the project depended upon interim forms of the architecture and exchange format. Nevertheless, valuable experience was achieved and useful principles established. It is sincerely to be hoped that this valuable resource is not to be wasted but may be applied in the further refinement of the GEHR architecture and to direct the evolution of prototype GEHR compliant systems and tools.

The GEHR consortium was able to demonstrate the comprehensiveness of the proposed GEHR architecture at the Audit presentations in Brussels in October 1994. The demonstrations are reported in Deliverables 16/17 [28] and showed prototype GEHR compliant systems and tools carefully integrated, functioning together and communicating effectively. The demonstration involved only a representative sample of what has been accomplished within the project, but still managed to convey the flexibility of the proposed architecture and its ability to span all systems from the very basic to the very sophisticated.

The results achieved thus far constitute a superb foundation upon which to build, and provide a secure base from which to launch into any subsequent phase. There is a need and a willingness to develop and expand the existing test sites, to extend their operation by collaboration with other European EHCR projects so as to achieve 'consensus, consolidation and standardisation' [29].

The 'Immediate Future Objectives' are summarised in the GEHR brochure produced to accompany the presentation in Lisbon, Portugal in December 1994. These include to:-

- Raise awareness of GEHR as providing a coherent basis for defining the kernel architecture.
- Ensure compliance with the emerging CEN EHR architecture standard.
- Disseminate the GEHR object model and exchange format in the public domain.
- Explore models of the EHR created in specialist clinical domains and map these to the GEHR object model and exchange format.
- Explore emerging health message protocols and map these to the GEHR object model and exchange format.
- Explore the interface of knowledge-base systems frameworks with the GEHR object model and exchange format.
- Explore emerging health care guidelines and map these to the GEHR object model and exchange format.

- Establish mechanisms for generation, evaluation and standardisation in appropriate domains of the fundamental GEHR record constructs: Transactions, Items, Collections, context attributes, etc.
- Establish a set of user test sites/demonstrators where GEHR compliance is specified, implemented and evaluated.

The objective is to use the feedback from the operation of the numerous test sites and the application of rigorous assessment methods, to redefine requirements and specifications, and to improve and refine the tools and later prototype systems [‘progressive refinement of the knowledge and the utility of standardisation of the structure, terminology and content of the EHCR’: GEHR Brochure].

The GEHR Support Action proposal [30] seeks to address these ‘future objectives’ and thereby maximise the extent and quality of feedback and to ensure that best use is made of this feedback. The intention is to continue fully to address user needs (and so increase acceptability). In the Support Action proposal, the user needs are of two kinds:-

1. The needs of all users interacting with the Electronic Healthcare Record (EHCR) - in particular the needs of the professional users, such as clinicians, managers, epidemiologists, assessors, clinical auditors, as well those needs imposed by the use of the EHCR in different Institutional settings. As the EHCR becomes more widely used, the additional needs of access by the public to their personal records, and the specific needs relating to Health Education will need to be met.

2. The needs both of Application System vendors and of users of EHCR systems in test sites for support while the EHCR Architecture undergoes a process of refinement. The main issues are:

- support for carrying out the validation process related to the use of the GEHR Architecture in the demonstrator sites;
- establishing and using two-way channels of communication between the demonstrator sites and the Support Action group for:
 - feedback of experience of using the Architecture to the Support Action group;
 - support for the test sites and vendors in further testing and adopting new versions of the Architecture.

Anticipating that several sites will be supported under Framework IV to validate and demonstrate Electronic Healthcare Record architectures, the Support Action seeks to be involved in helping to draft the components of the validation methodology which specifically address EHCR architectural issues.

The purpose of the proposed action is to assist progress towards a better final architecture and tools which are effective and user friendly: 'Good health records - good in the sense that they directly improve the quality of care given to patients and good in the sense that they provide a comprehensive and portable record which can fulfil local, national and international professional, ethical and legal requirements'. In other words, the intention is:

- That a complete architecture and exchange format will be available for an EHCR, supported by a library of drawings, to enable system developers (vendors) to provide users with good quality EHCR systems.

- That users be encouraged to keep good quality EHCRs and be able to share them with others.
- Systems based on the GEHR Architecture will enable monitoring of clinical care by individuals and by teams of clinicians, subject to ethical constraints. This in turn will facilitate audit and lead to more effective care.
- Continuity of care and the integration of specialist and generalist perspectives within the record will lead to a comprehensive record. This, in turn, could form a valuable resource for health education and disease prevention, both for individual patients, and, in aggregated form, for larger populations.
- That patients should perceive an increase in the quality of their care, because of more efficient shared care between health sectors. (e.g reduction in unnecessary tests, visits because of the possibility of General Practitioners' having Hospital data in their EHCRs and vice versa).

The proposal will provide support for projects which seek to demonstrate that the wide adoption of electronic healthcare records within a diversity of healthcare enterprises is both feasible and beneficial. By adopting common methodologies and extensive validation programmes these supported projects hope to stimulate the development of clinically acceptable and cost-effective healthcare records, multi-media and communications products, backed up with a comprehensive user educational programme.

EHCR Application systems are already available in partially GEHR-compatible form. These products, and others coming on the market in the next few years will be increasingly compliant with the Architecture. It is envisaged that some products will be fully compliant by the end of the first year and will continue to improve, as the Architecture itself is refined.

10.1.1 Anticipated Areas of Future Development

The GEHR Architecture provides a framework which supports the full diversity of clinical data storage and communication requirements. It is formulated to encompass the different disciplines of primary and secondary healthcare, for doctors, nurses, and other professionals and in all European countries. The ready access to a wide range of data-types is of increasing clinical importance, and the work of the project has already included these multi-media aspects of the record architecture.

The expected rapid growth in the availability of high band-width communications technology will inevitably create an expectation for interchanging or exchanging large volumes of data routinely. GEHR has already foreseen this trend and proposes a generic approach to the handling of bulky EHCR data.

Further exploration in the test sites of the realistic use of Multimedia data in real EHCR applications will take place, and the experience fed back to a follow-on GEHR initiative. Any necessary Architectural features discovered to be necessary will be added to the base Architecture.

It has been and continues to be important to make the Architecture open to progressive refinement and to the incorporation of new requirements and advances, such as in the systematisation of medical knowledge and in the utilisation of new technologies.

The Support Action Proposal recognises the need for an ongoing propagation and evolution of the Architecture in addition to any demonstrator site evaluations.

Similar considerations apply to the expression of the model itself. The GEHR model is currently described diagrammatically (using a Rumbaugh style notation) and textually in the programming language Eiffel. Neither of these two descriptions is formal in the strictly mathematical sense. The benefit of using a mathematically formal description is that the primitive objects in such a description would be mathematical objects such as sets and functions together with first order logic. This would mean that any detailed questions of meaning in the GEHR model could be reduced to statements about mathematics.

The existing GEHR model description would serve as a good basis for the construction of a completely formal description. There are a few object oriented specification notations such as Object-Z and Z++ which seem to be suitable given the object oriented nature of the model. In practice, the construction of a formal description usually exposes subtle ambiguities in the informal description and these would have to be resolved by the designers.

10.1.2 Future Standardisation: Relationship between GEHR and CEN

Innovation precedes the confirmation of standards in any field, and GEHR has expressly sought to share its work with interested parties at an early stage. The GEHR Deliverables have been shared with other AIM projects and with CEN, and the members of the Consortium present on CEN working groups have sought to facilitate a common perspective.

The interim specifications of the GEHR Architecture have formed a major input for consideration by PT011 of CEN TC/251/WG1, and in particular towards the First Working Document of PT011 on the standard for the Electronic Healthcare Record Architecture (EHCRA).

Proposals are being considered by TC/251 for the establishment of a Project Team to work on a standard for an Extended Architecture (EHCREA), and for other related activities.

The later work of GEHR in specifying higher level structures of the Architecture should form an important input to this new work of CEN.

Any support and maintenance project which involves the continuing development of the GEHR architecture must maintain an active working relationship with the standards makers. The objectives of any such project would include:

- To ensure that the GEHR work and experience informs the CEN TC251 EHCRA and later standards;
- To propose methods for ensuring that the continued promotion and refinement of the GEHR Architecture can continue after the end of the Framework IV period, suitably harmonised with the CEN plans for future EHCRA standardisation.

The principal activities of such a project would be to:

- Seek representation on the appropriate Working Groups and Project Teams of CEN TC/251;
- Disseminate the GEHR products and materials to CEN to inform the work of Project Teams;
- Provide proposals to CEN for improvements to the Electronic Healthcare Record (Extended) Architecture and associated standards as they evolve through PT011 and its successors;
- Propose future support/maintenance mechanism (taking into account what CEN will have as its EHCRA support mechanism);
- Set up and use technical methods for testing the GEHR compliance of application systems. Issue “seal of approval” to compliant systems.

Note: this might later turn into some kind of “approval” operation, for which a charge could be made, and put towards the continuing cost of running a future support organisation such as that described under ‘GEHR Authority’ below.

Internal Project Deliverables might include:

- Membership of PT(s);
- References to GEHR materials in EHCRA and later standards;
- Documented compliance testing procedures;
- Plan for continuing maintenance of the GEHR Architecture.

The Quality Plan for such a project would prescribe the use of various QA procedures and evaluation techniques, feedback from which will help to refine not only the GEHR architecture, but also the evaluation / testing procedures themselves. Utilisation of such methods and adherence to the project standards (as evidenced by compliance checks) will ensure that the documents and other Deliverables produced will be of an appropriate quality to feed into the standardisation and other activities.

Final Deliverables of a support project might include:

- Report of liaison with CEN, compliance procedures and testing.
- Final report of liaison with CEN; statement of comparison between the latest EHCR standards from CEN and GEHR.

10.1.3 GEHR Authority

The section above indicates the importance of longer term issues, the continuing requirement to monitor the work of influential bodies and authorities. For example, CEN is offering a structure which identifies registered coding schemes - the HCD (Health Care Coding Scheme Designator). It is also (via TC 251, Working Group 3.8) looking at the interchange of health care information. A scheme for the Registration of Coding Systems (RCS) is proposed, to take account of existing ISO and other standards and to constitute a standard to cover the process of registration and

maintenance of coding lists. A similar scheme is proposed for the recording of data sets (WG 3.9).

These and other longer term considerations have important implications for GEHR. It is likely that some kind of GEHR foundation or controlling body/licensing authority will be required to manage such issues as:

- licensing GEHR compliant systems and performing conformance testing;
- maintaining the GEHR standard + handling upgrades:
 - authorising changes and handling change control;
 - monitoring and ensuring compliance with related standards (existing and developing);
- registering and supporting users:-
 - training of users,
 - support for user groups,
 - technical support;

Many of these responsibilities fall under the umbrella of 'maintenance'. The essential requirement, as with maintenance during the project, is to have effective and efficient control of all these processes. The requirements for an effective change control body have already been covered in GEHR Deliverable 23 [31]. Briefly, maintenance management must be undertaken by a responsible body with clearly defined authority. It must have clear objectives and be subject to strict quality control itself. The fundamental mechanism is change control. Thus, it is necessary to have:

1. A change control body: to consider change requests and authorise or disapprove the changes.
2. A responsible maintenance team: to implement changes/upgrades.

Successful maintenance management involves:

1. planning;
2. monitoring and control;
3. organising and implementation;
4. human factors.

In fact, these are exactly the kinds of characteristics which are required for the effective and efficient handling of feedback generated by any follow on project.

The list of potential responsibilities of any GEHR authority is not exhaustive by any means. It does, however, serve to illustrate the breadth of the duties involved. For example, the control of changes and GEHR versions must be handled so as to preserve some kind of compatibility with earlier versions. Upgrades of term sets will have to be validated. Any GEHR authority will be the ultimate arbiter in conformance testing to establish the degree of compliance of any particular system with a given GEHR standard. In this respect, an OO model such as the GOM can be very accommodating, allowing differing conformance (or compliance) levels.

Such responsibilities are not limited to the architecture. The requirements of maintenance dictate that the same GEHR authority will have to monitor developments in associated areas and ensure that GEHR keeps pace and maintains compatibility with current and emerging industry standards

and protocols. For example, the project has built and evolved specifications for tools to interface between the object model and the growing range of standard messages emerging - e.g. laboratory data (EUCLIDES, OPENLABS), images (DICOM), ECG (SCP-ECG), and generic messages such as HL7 and message protocols such as EDI. GEHR must be able to evolve so as to keep pace with changes in these fields, thereby maintaining its place at the leading edge of technology.

Any eventual licensing authority will be responsible for supporting users as well as administering the operation of GEHR compliant systems. Users will have to be registered/licensed to use GEHR systems. They will then become entitled to full user support, membership of user groups, advice via technical support and training schemes and, not least, upgrades as they come along.

11 References

1. 'Good European Health Record - GEHR Project A2014' - brochure produced for the AIM Conference, Lisbon, Portugal, December 1994, St. Bartholomew's Hospital Medical College.
2. 'GEHR Object Architecture - Recommendations', Version 2.1 Draft A, Tom Beale, St. Bartholomew's Hospital Medical College, 25/11/94.
3. GEHR Deliverable 10: 'General Syntax and Semantics - Interim Description'. 1994
4. 'Data Analysis, Data Modelling, and Classification', M. Modell, New York: McGraw-Hill, 1992.
5. GEHR Deliverable 4: 'GEHR requirements for Clinical Comprehensiveness.', 1992.
6. 'Requirements for Comprehensiveness in Hospitals and across Specialties', A. P. Maskens, Health Data Management Partners, Brussels, Belgium, Dec. 1992.
7. GEHR Deliverable 7: 'Clinical Functional Specifications', 1993.
8. GEHR Deliverable 5: 'GEHR Requirements for Portability'. 1993
9. GEHR Deliverable 6: 'GEHR Requirements for Communication'. 1993
10. GEHR Deliverable 8: 'Ethical and Legal Requirements of GEHR Architecture and Systems'. 1994
11. GEHR Deliverable 9: 'Educational Requirements of GEHR Architecture and Systems'. 1994
12. GEHR Deliverable 18: 'GEHR Software Tools: Final Specification', 1995.
13. GEHR Deliverable 15. 'Final GEHR Architecture Report', 1994.
14. 'Object-oriented Software Construction', Bertand Meyer, Prentice Hall 1988 (2nd Ed. 1994). [the basic text in OO theory]
15. 'Seamless Object-Oriented Software Architecture', Kim Walden & Jean-Marc Nerson, Prentice Hall, Sept 1994. [documents the BON method, an analysis level modelling formalism for seamless implementation]
16. 'Object-Oriented Modelling', James Rumbaugh et al., Prentice-Hall 1991. [an Entity-Relationship approach to OO Modelling].
17. 'Object-Oriented Design with examples', Grady Booch 2nd Ed., Benjamin/Cummings, 1991. [a C++ - implementation-based approach to OO design].
18. 'Eiffel The Language', Bertand Meyer, Prentice Hall 1992. [reference for the Eiffel language used as the reference language in this document]
19. 'Re-usable Software: the Base Object-Oriented Component Libraries', Bertand Meyer, Prentice Hall, 1994. [a treatise on software library construction, offering an actual set of library interfaces to be implemented in any language. Proposes Linnaean-style classification, consistent naming, and correctness as methods.]
20. 'The GEHR Object Model', Version 2.2, T. Beale, D. S. L. Lloyd. Internal GEHR Project Document. St. Bartholomew's Hospital Medical College, 1995.
21. 'Object Management Architecture Guide', R. Soley (ed.), Object Management Group 1991.
22. 'The Common Object Request Broker: Architecture and Specification', OMG, 1991; OMG Document 91.12.1
23. The Object Database Standard (ODMG-93), R. G. G. Cattell, Morgan Kauffmann 1993.
24. 'The GEHR Systems Architecture Guide', T. Beale, St. Bartholomew's Hospital Medical College. 1994.
25. 'Distributed Object Management', Morgan Kauffmann 1994.
26. 'GEHR Interim Software Tools - Interim Specification', Deliverable: [11], University of Hull, 1994.
27. GEHR Deliverable 13: 'Final Systems Report', University of Hull, 1994.

28. GEHR Deliverable 16/17: 'GEHR Software Tools and Prototypes / Application Software Integration', D. A. Camplin, University of Hull, 1994.
29. 'Position Statement and The Way Forward for GEHR', D. Ingram, D. Kalra, D.S.L. Lloyd, 1994.
30. GEHR Support Action Proposal (Framework IV), D.S.L. Lloyd, March 1995.
31. GEHR Deliverable 23: Documentation & Maintenance Procedures, December 94.
32. 'Universal Resource Identifiers in WWW', Tim Berners-Lee, ("http://info.cern.ch/hypertext/WWW/Addressing/URL/URI_Overview.html")

Appendices

A Information and Object Modelling

The definition of an information model is largely unrelated to the technical problem of transmitting information conforming to it. For instance, certain attributes of EHCRs such as who committed a Transaction, date and time of commitment, etc., are required not to facilitate transmission in any way, but to satisfy medico-legal requirements. These attributes must exist regardless of whether the information is transmitted anywhere else. Other requirements on the structure of the information are related to making it comprehensible to other health care professionals, via whatever applications they have available to them to process it. There may be attributes required which relate to the fact of movement of information among sites, but again, these are separate from *how* it is actually moved. It is therefore possible to define the information model independently of an exchange format.

Any model proposed by the GEHR architecture must be, by definition, a model of “standard” underlying information structures, and *not a model of any particular view* of such information. This is because in a wide context (e.g. a hospital, a country, Europe) there is a multitude of users using different applications, data storage mechanisms, and computer platforms. The applications will reflect many views and uses of information. Since it is not the remit of GEHR to legislate on applications or in fact any aspect of implementation, the only possible subject of a GEHR standard information model is the underlying information itself.

It is therefore important to understand the structure of the underlying information, rather than that of a view of it. By common consensus, it is likely to consist of cross-linked n-dimensional hierarchies of Headings, Collections, multi-media Items, text and so on.

There is a number of important consequences of these observations:

- familiar ways of thinking about the information such as the hierarchically indented linear lists of Headings and Items written by practitioners must be used with caution: these are only views of the information as seen by particular practitioners, and are necessarily simplistic due to the nature of the medium. A consistent semantic model of underlying information cannot be developed by using such views as if they were true representations of the information.
- there is an infinity of ways that the same underlying information structure could be created. Assuming the use of interactive GUI applications by many users, the underlying information will often be constructed by human input through dialogue boxes, menus, text entry, etc.
- no human will ever “see” the information itself, - it will only be experienced through views generated by mechanisms (such as computer applications) capable of manipulating it;
- there is an infinite number of particular ways a given piece of complex information could be viewed or experienced by people, and a multitude of basic formats in which such views could be experienced, including:
 - as a hypertext multi-media presentation;
 - through an interactive program;
 - as a printed document;

- as tabulated query results;
- in a spreadsheet-style program;
- translations in other languages of terms and free-text;
- as synthesised information such as graphs, summary reports etc.
- the way a data storage mechanism at a site “sees” information constitutes yet another view of the information.
- The only sensible way to exchange primary information is to *transmit the underlying information itself*.

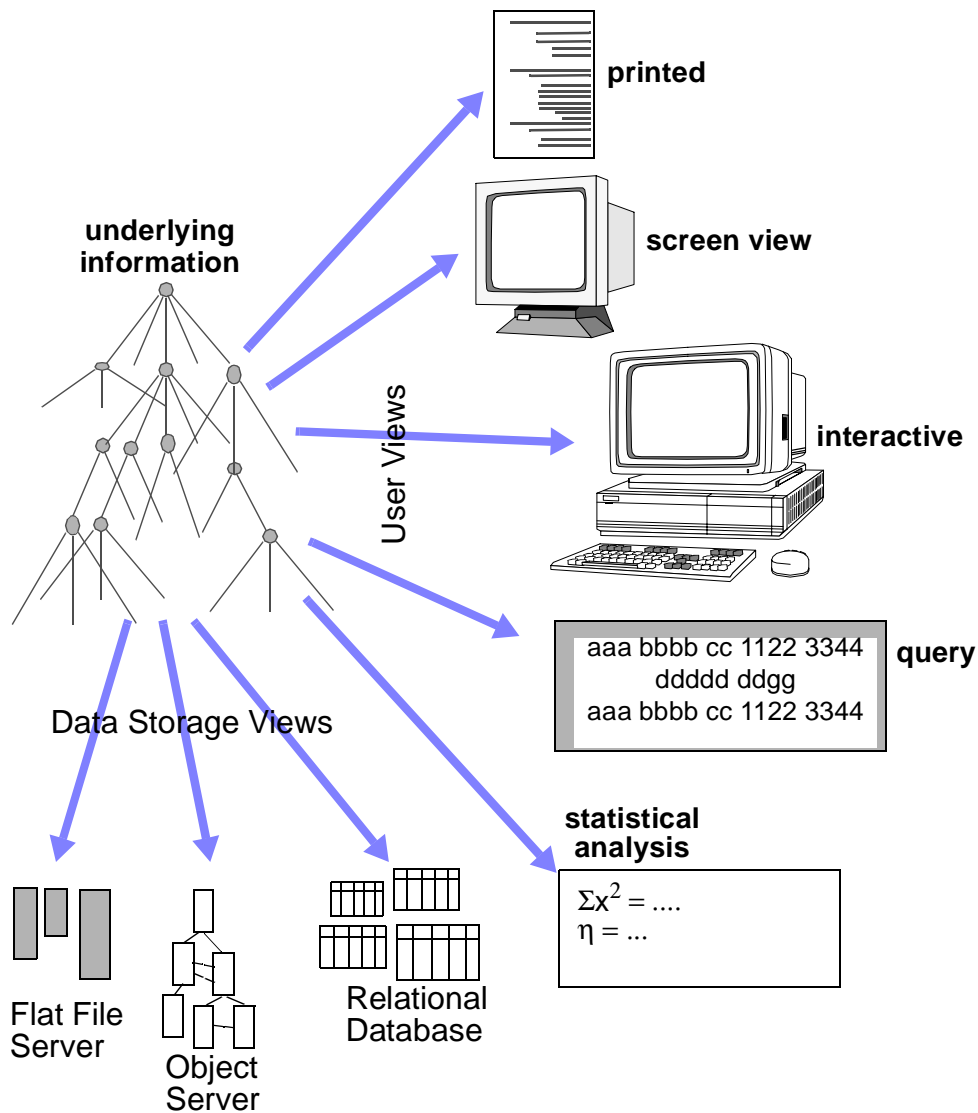


FIGURE 22 Underlying Information vs Views

These comments should not be seen as minimising the importance of information views, since it is via views that practitioners will record the practice of health care. What is being suggested is that to achieve a flexible system whose information structure and content satisfies certain requirements, the underlying information must be modelled. In fact, it is this approach which will give implementors the greatest freedom since the design of views of information is not being prescribed.

A.1 Requirements of a Formal Model

A.1.1 Validateable

The first and foremost requirement for any model which is going to be used (rather than remain a topic for theoretical discussion) is that it is validateable. Otherwise it would be possible to define a model which is not even self-consistent in its own semantics. While it is, in theory, possible to validate a formal model by hand (e.g. using propositional logic or some such device), in practice a formalism must be chosen for which building and validation tools exist. It is for this reason that a model such as that proposed in Chapter 6 should never be used directly as a formal model definition. The model is only described and illustrated, not actually expressed in a form readable by a tool.

A.1.2 Definition of Ideas

The primary use of a model is to capture the thinking about a problem in such a way that anyone literate in the formalism can understand, that is to express a *definition*. Mathematics and Logic are two well known formalisms in global use which serve exactly this purpose. As already stated, a rigorous formalism can prevent ambiguity and confusion inherent in the use of natural language. It can also encapsulate in a disciplined way the knowledge agreed upon by the users of the formalism.

Currently published structural and object-oriented modelling “methodologies” are essentially modelling formalisms of varying degrees of rigour.

For the purposes of defining a GEHR architecture, a formal object (c.f. structural, procedural) model is extremely desirable, since:

- the complexity of medical information lends itself to a semantically powerful model such as an object model;
- it gives the ability to state agreed-upon ideas in unambiguous terms;
- object models are accommodating of functional and structural ideas;

A.1.3 Communication of Ideas

The second main use of a formalism is to facilitate the communication of ideas. In the same way that any mathematician in the world can read and understand the equations of another mathematician, regardless of their mother tongues or cultural backgrounds, the semantics encapsulated in a formal model can be understood by any reader familiar with the modelling formalism.

Since the development and usage of the GEHR architecture is in a Europe-wide context, the use of a formal model as a communication mechanism with the ability to transcend (at least some of) the problems of natural language descriptions is crucial.

A second aspect to communication is the distribution of the model itself. While a “paper” version (e.g. an extracted document) could be distributed in order to communicate the ideas, a formal version is needed in order to distribute a usable form. Thus, the formalism chosen must be processable by every party involved, using tools conforming to the definition of the formalism.

A.1.4 Implementable

As with any problem for which computers are used in the solution, the formal model is also the basis for implementation. Not only must the model be comprehensible by modellers (analysts), it must be comprehensible to implementors as well. A formal model should be an indication that the model is in fact implementable at all, since an unimplementable model is useless.

It is here that the concerns of heterogeneous systems must be addressed. As previously suggested, it is essential to create a model which allows multiple levels of conformance, and therefore interoperability of diverse information resources.

A.1.5 Conformance Testing

A powerful formal model can be used much more easily to determine conformance of implementations to a required standard than can a natural language, or even a semi-formal description.

If multiple levels of conformance are defined in the model, this should be done as part of the formalism if possible.

A.1.6 Completeness

It is of great importance to achieve two things with the choice of modelling formalism:

- the modelling formalism must have the semantic power to represent every concept required to describe adequately the information in the problem domain. If this is not true, the model has to be completed with natural language, rule definitions or some other mechanism to provide fully the semantics. Concepts such as aggregation and inheritance are for example handled by an object modelling formalism, but not by an EBNF one;
- it should be possible to include every entity in the problem domain in the model, ensuring a minimum of contention due to informal definitions. For example, if the concepts of EHCR and TRANSACTION exist, they need to be expressible in the model.

This is not to say that the informal description should not be used to explain and aid the understanding of the reader.

A.2 Object Models

A.2.1 Semantics Available in an Object Model (OM)

Detailed descriptions of the main elements of Object-Oriented formalisms may be found in [14], and [15], [16], and [17] as well as many other publications. The following provides a brief summary:

The principal construct of an object model is the class, which encapsulates both attributes and routines (state and behaviour) as features. Class definitions define the type of instances created at runtime; often there is a 1:1 correspondence. Inheritance provides the capability to define classes which inherit and possibly change in restricted ways, the features of a parent class. Inheritance leads to the powerful concept of polymorphism: instances of subclasses are by definition also instances of all parent classes, and can therefore be handled at runtime without knowing or caring about their actual type. Generic classes may be defined to handle instances of any particular set of classes as arguments, for instance LIST[G], SET[G], and TABLE[G, H->HASHABLE] define abstractions which will take other classes as arguments.

In some OO formalisms, correctness can be added to a model at both the routine and the class level. Routine pre- and post-conditions expressed as logic statements define entry and exit correctness conditions for executing the routine; class invariants define correctness condition true at all times for all instances of the class.

Most Object modelling formalisms provide for various kinds of association or runtime relationship between instances. Aggregation is a particular form of association whose meaning is “part/sub-part”, and which is usually distinguished by propagation of operations through the sub-part hierarchy.

The relationships of inheritance and aggregation are ubiquitous in most object models, and likely to occur extensively in a health care information model, so some further comments are in order. If the concept of “jet aircraft” was to be modelled from the point of view of say an air-traffic control system, the model might look as in the figure below.

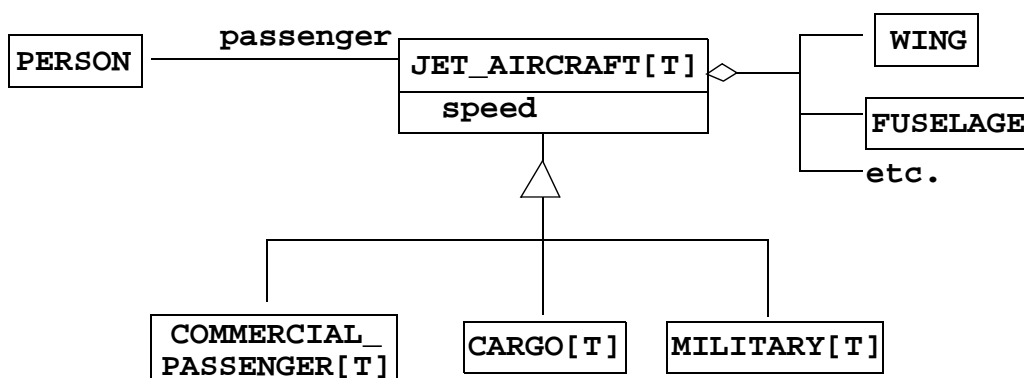


FIGURE 23 Inheritance, Association and Aggregation Example

Three kinds of relationship are used:

- inheritance: 3 kinds of aircraft are modelled, by inheriting from the abstract concept of JET_AIRCRAFT;
- aggregation: the construction of a JET_AIRCRAFT is modelled by the aggregation relationships with WING and FUSELAGE;
- association: the generic class JET_AIRCRAFT has an association called passenger with the PERSON class;

Imagine an instance of JET_AIRCRAFT, representing a real aircraft in flight. The speed attribute will be set to some value, e.g. 780 km/h. Obviously the wings and fuselage of the aircraft, and the passengers are also flying at this speed, since they are all attached to the aircraft in some way. But although they appear to “inherit” the speed attribute, they are not related by inheritance: neither WING nor PERSON are kinds of JET_AIRCRAFT. What is in fact happening is an contextual propagation of a feature due to aggregation (WING and FUSELAGE) or containment (PERSON). Thus although none of these other classes have a speed attribute, the query “what is the speed of a person in the aircraft?” is still meaningful, for the time that the person is in the aircraft.

Propagation of this kind occurs with containment or aggregation. Explicit propagation of a feature can also be achieved by defining the feature on the required part and sub-part classes. For example, a feature position could be defined for JET_AIRCRAFT, and for PERSON. Then for PERSON instances inside an aircraft, the position feature would be explicitly propagated from that of the aircraft.

A.2.2 Seamlessness

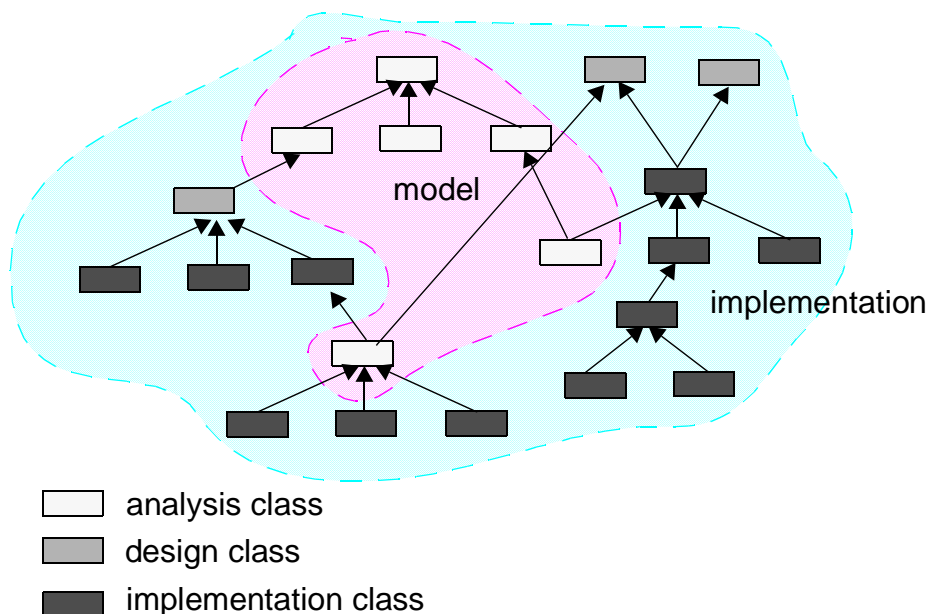


FIGURE 24 Object Model Seamlessness

An object model is seamless at all levels of abstraction, since only one type of construct is used: the class. An analysis OM may be extended at the design level simply by adding more classes. No new construct is required. The same is true for the implementation level. A complete OM at the implementation level actually consists of:

- analysis classes (e.g. PERSON, EHCR_ITEM)
- design classes (e.g. SET, LIST, ITERATOR)
- implementation classes (e.g. LINKED_LIST_ELEMENT)
- code created during implementation (e.g. routine bodies) in all classes

The importance of this is that an object modelling formalism is also appropriate for implementation. This is not true of some other modelling formalisms (e.g. EBNF). Stated in another way, the model and the software are one and the same thing.

A.2.3 Extendible at any Level

One of the most useful properties of an OM is that it is seamlessly extendible. This is due to the use of inheritance: if changes are required at some later point in time, they can often be incorporated in such a way as not to invalidate any previous definition or implementation of the model. Changes are made according to the rules of conformance (see [18] for a precise definition) which include the following:

- new classes may be added, including classes which inherit from existing classes. As an example, imagine that in a PATIENT class hierarchy, a new type of PATIENT is required in order to model cancer patients. To upgrade the model, a new class CANCER_PATIENT is defined and added to the inheritance hierarchy. The change does not invalidate the previous version of the model (all previous semantics are intact) and does not harm implementations (those based on the previous version of the model are compatible with new implementations, but do not deal with all patient types);
- new features (attributes, functions, procedures) may be added;
- existing features may be changed in a conformant way as determined by the inheritance relation (see e.g. [18] for one precise definition), which entails:
 - features may be renamed;
 - number and order of arguments to functions and procedures may not change;
 - wherever types occur, they may be changed to any subtype. For example, name:PLAIN_TEXT in class X could be altered to name:TERM_REF in class Y, a descendent of X, if TERM_REF inherited from PLAIN_TEXT;
- correctness conditions can be changed in a conformant way. Roughly this includes:
 - weakening of routine pre-conditions;
 - strengthening of routine post-conditions;
 - or-ing of invariants.

A.2.4 Naming

The problem of names of elements in a formal model is of great importance, especially in OO since an OM purports to be an expression of ideas at a semantic level very close to that of normal thinking. Naming problems are exacerbated when there are multiple contributors and multiple

mother tongues, as in the GEHR project. The problems of naming are dealt with in many books, including [19] (a novel and powerful approach) and [16]. Some basic principles include:

- care must be taken to choose words not already having a “loaded” meaning from any of the disciplines involved in the project. For example the words “transaction” and “record” as used in GEHR are both well-defined concepts from the database and OLTP worlds, and care must be taken to see that their original meanings do not obscure the intended meaning;
- terms should be chosen at the right level of generality. For example, a term like “collection” is likely to appear at a high level (i.e. near the root class) in any OO class library, and is likely to be associated with some very generic characteristics;
- systematic naming as proposed in [19] is recommended, as it improves the power of inheritance, and simplifies understanding the abstract properties of modelled entities;
- whatever names are chosen, their precise meanings must be defined in the model.

A.2.5 Modelling Incomplete Concepts

It is often desirable to include in a model some idea of concepts which are not yet fully understood. How can this be achieved within a formal object model? Quite simply, given that everything which is included in an OM is potentially incomplete. Even the concepts whose models are thought to be complete are revised as requirements change and users provide feedback. In fact, the entirety of an OM is potentially incomplete. There are, however, a few simple guidelines for thinking about concepts which are known to be incomplete at the time of modelling:

- if there is a strong problem domain concept for which the semantics are unsure or which cannot be differentiated yet from something else already modelled, include it as a separate class. This provides a place where semantics can be attached when they become known; if it is later discovered that it is identical to some other class, or completely superfluous, it may usually be removed from both OM and the various views (especially implementations) easily. Belatedly introducing classes however, is often quite difficult, and is not recommended.
- the opposite is true for features in classes: it is better to avoid the tendency to add every feature that “might one day be used” and keep to those for which a requirement exists. Excessive numbers of vaguely defined features simply add noise to the model, and obscure the strong concepts.
- semantics of correctness and functionality can be defined in their own right in a model. For example the NUMERIC and COMPARABLE classes in the Eiffel libraries ([19]) model various capabilities without introducing any kind of data representation. The use of pre- and post-conditions and class invariants as proposed in [14] and [18] is very powerful here.

A.3 General Approach to Modelling

A.3.1 Comments on Modelling Formalism Chosen

The formalism chosen for this document consists of:

- diagrams based on the “Rumbaugh” methodology ([16]), with concepts added from the Eiffel language ([18]) and the BON notation ([15]);
- an underlying textual representation expressed in the Eiffel language;

Although Rumbaugh diagrams are not truly formal, and do not deal with all of the possible semantics of an OM, such as correctness, they do not prevent the addition of extra semantics to classes and operations. Rumbaugh has been chosen as a basis in the interim, as it is widely known and used, and is not flavoured too heavily by implementation considerations.

In terms of the requirements for a modelling formalism, it may be rated as follows:

validateable:partial.

Some level of validation exists in most Rumbaugh tools. Since there is no formal grammar for a Rumbaugh model, validation is implementation dependent. Note: for this work, no Rumbaugh tool is being used, so this validation is not available;

definition:quite effective.

Rumbaugh diagrams are effectively Entity Relation diagrams with OO additions, although the implied semantics are OO. There are many CASE tools which implement the Rumbaugh method for constructing models;

communication of ideas:good.

Rumbaugh has been available since 1991 and has good acceptance; the diagrams are mostly intuitive;

implementable:reasonable but not seamless.

Many CASE tools can generate C++ or Smalltalk code skeletons, but no reversibility is implied. Since there is no explicit formal grammar for Rumbaugh, CASE tools use whatever internal representations they wish, and there is no seamless way to map into an OO programming language;

compliance testing:given that there is no formal grammar, there is probably no watertight method that could be applied to an implementation to determine whether it was in fact compliant;

semantic power:reasonable at a basic level, but: no correctness semantics, e.g. operation pre- and post-conditions, class invariants; no generic classes;

Under this scheme, the Rumbaugh diagrams are created “by-hand” and then coded in Eiffel, which actually expresses the model formally. Corrections are fed back into the diagrams. In future it is recommended that a formalism be adopted whose text and diagrammatic representations are completely seamless, and for which building and validation tools exist.

Possibilities include:

- Eiffel + the BON method graphical modelling language (formal grammar seamless with Eiffel (see [15]);
- Object-Oriented Z;
- the OMG Object Architecture (a formal abstract grammar) (see [21]);
- the OMG IDL syntax (a formal concrete grammar) (see [22]);
- the ODMG object model (a formal grammar; describes a superset of the OMG OA) (see [23]);
- formal standards including ISO ODP, ANSI X3H7 (Object Information Systems) (status unknown);
- OSF/DCE CDL (equivalent to OMG/IDL);

It is also essential that the actual grammar or interpretable diagram format be used for validation and implementation. Further investigation into the advantages and disadvantages of these formalisms is required before a final choice can be made.

Pending any such study, the BON/Eiffel path is proposed, since BON and Eiffel are effectively the same language, with a fully defined grammar and diagram equivalents, and are available in the public domain. They are also semantically superior to other choices currently known, which means they can be “mapped down” onto other formalisms. Furthermore, a reasonable selection of tools is currently available.

1

A.3.2 Minimum Features

The basic approach taken in the model is to define the minimum classes and features which could be standardised across Europe in order to satisfy the clinical and ethico-legal requirements in [5] & [10]. The nature of OO models means that the GOM can then be formally extended in a number of ways to create database schemas, software libraries, exchange definitions and other views, in both OO and non-OO technologies.

Thus the model represents what is necessary as a minimum, with the intention that it be extended at each site to fully model and implement what is required. Details of how to transform and extend the model are given in [24].

A.3.3 Function Features

Functional features have been used in some places in the model, in order to capture precise semantics which happen to require arguments. This often causes confusion in models written in a formalism which, like Eiffel, is also compileable to an implementation. It should always be remembered that they appear in the model as formalised semantics for things like actions or queries which depend on external information (hence the need for parameters) and are making no statement whatever about implementation.

A.3.4 Correctness

Many semantics in the model are expressed as logical conditions. The Eiffel language directly supports correctness conditions (including in compiled applications if desired) expressed in first-order predicate logic statements of three types:

- **class invariants:**
condition must always be true in every instance of this class
- **routine pre-conditions:**
condition must be true for correct functioning of routine;
- **routine post-conditions:**
condition will be true on exit of routine (assuming the pre-conditions were met).

The design of the correctness features in Eiffel is based on the Z specification language and the concept of “programming by contract”. See [14], [18], and [15] for detailed discussion.

A.3.5 Classification of Features

Class features are classified in a number of ways, in order to clarify the purpose in mind. Of the large number of classifications, the following are important and occur commonly:

- Identification: features which identify instances of the class;
- Context: time and place context of information stored in other features of the class;
- Content: the information content of instances of the class.

A.3.6 Where does the Model Stop?

All classes defined in the model are by definition internal to the model, and its derived views. The model can be thought of as “interfacing” with the external world via external references, which refer to entities which the model understands in a generic way. Examples in the GOM can be found in the HCP, HCF, and EHCR_INFO classes.

A.4 Object References

The OO formalism used for the GEHR Object Model includes the concept of “object reference”. This Section attempts to explain the meaning and implementation of references, and to reduce some of the confusion between the logical reference concept and implementation concepts such as “pointer” and “address”. The role of unique identifiers (UIDs) is also discussed. For more detailed discussion, refer to e.g. [14], [18], [22], [25].

A.4.1 Meaning

In an object model, instances can be directly associated with each other in only two ways (as previously mentioned):

- logical reference
- “inline” expansion

The vast majority of instance associations are of the first kind, the second usually being reserved for optimizing access of objects such as integers, characters, booleans and so on. For example, in the Eiffel system, only five classes are normally “expanded” like this (although in fact any class can be expanded if necessary). A reference is thus a run-time association between two objects.

Logical references are often confused with various kinds of physical reference; in fact no particular implementation is implied by references in an object model. The only implementation-level semantics implied by logical references are that they are distinguishable from all others, i.e. uniqueness.

Indirect references between objects are also possible, as described in the previous Section, and it is this type of reference that defines the limits of an object model. For example, an object may refer to a file, something in a database, a remote object or even another local object by using an external reference rather than a direct reference.

A.4.2 Implementation

The implementation of logical object references may change during the lifetime of the object. When an object instance is created, references are usually implemented with memory addresses, since this is what most compilers will generate for von Neumann machines. If the object’s lifetime is less than or the same as that of the program, and it never moves, its references will remain as addresses, leading to the misconception that they are the same thing. However, there are (at least) three ways this might change:

1. if the object is made persistent, e.g. stored in a file or database;
2. if the object is moved from its original address space in the same context, e.g. transmitted to another process managed by the same database system;
3. if the object is moved beyond its original context, e.g. to another database or system.

The term “context” is defined as the scope in which the object is uniquely identifiable without further qualification. In both cases, two problems arise:

1. validity of references: in-memory pointers will no longer be valid when an object is stored or transmitted, since addresses only retain the same (effective) value during the same execution, due to memory management by the operating system;
2. referential completeness: all or some subset of the objects connected by references must be dealt with. There are two possibilities:
 - follow the references until all connected objects are found and treat them all similarly;
 - move only a subset of the graph of connected objects and either convert the “hanging” references to a form which retains their meaning (i.e. which objects they were pointing to) or else ignore them and lose some proportion of the referential integrity of the object graph.

These problems provide the motivation for changing the implementation of references either on the fly, or by design, as described below.

A.4.3 Persistent Objects

If an object is made persistent, which means to extend its lifetime beyond that of the system execution, its references must be converted on saving to a form which is valid during storage (usually in a file or database system), and which can be converted back to the original form when the object is brought back into the execution environment. A typical solution in object-oriented systems is to replace the addresses with Unique Object Identifiers (OIDs or UIDs), each of which is unambiguously associated with a single object instance. UIDs are usually large integers or bit patterns whose structure is optimized for fast retrieval. The use of UIDs presents two problems:

- each object needing a UID must have one created or generated for it, the value of which never changes during the entire life of the object (otherwise referencing objects may become unwittingly invalid);
- UIDs are only valid within the context in which they are defined, which may be for example a certain host or a database server extending over multiple hosts.

A.4.4 Exchange of Objects

Moving an object to another location within the context but outside the address space is another circumstance forcing conversion of addresses. If the receiver of the object was a process in which objects were represented in exactly the same way as the sender, the UID scheme could be used, and without further modification, the binary image of the object could be sent. In the more general case, however, different programming languages, and even different platforms and hardware architectures should be assumed. In this case, the entire object must be converted to a format that both parties agree on, and which preserves references. This problem is the reason for the existence of such systems as RPC, CORBA, DCE and ASN.1. In these systems, objects are “serialised” into a byte stream for transmission and “deserialised” by the receiver. References may be converted to UIDs, or other schemes such as byte offset from the stream start may be used for efficiency.

When an object is transmitted beyond the context in which UIDs are valid, references to objects remaining behind have to be handled differently to retain validity. The main difference is that an identifier representing the context must be logically prepended to the UIDs. This is where addressing schemes such as those encapsulated in the WWW URI scheme are used, including ftp, wais, gopher and http to name a few.

A.4.5 Persistent and Exchangeable Objects

An object model describing potentially persistent or exchangeable objects may have to take into account the above considerations to be valid, as follows:

persistence:

Objects being made locally persistent need UIDs defined and implemented. All applications at a site which use the same storage mechanism, regardless of programming language, must have same understanding of the persistent reference scheme, since it is imposed by the database system, not the applications. However, it does not matter that each site uses a different scheme, since the only “clients” of the persistent objects are site-specific applications (or more usually, a dedicated database server).

Since we are not standardising databases at sites, we do not need to standardise persistence reference mechanisms either.

exchange:

During the process of exchange between sites, both parties need to have a standard idea of the UID or other object identification system. For the case where all objects connected by references travel together, this problem exists only during the period of transmission, and is dealt with in the exchange binding for the programming language (which defines how to convert a “reference” to an ID of some kind to be used during transmission).

This is satisfactory for objects travelling together, but in the situation where some objects are left behind, references have to be either voided, or substituted with remote references which then remain intact. In this case, the OM must include the concept of remote references for all object types which will have potentially to reference other objects remotely due to copying or moving.

external objects (active and passive):

These are “objects” outside the context of the referencing object but usually within the same computer system: their semantics are unknown to the model, and are therefore handled opaquely. For example, records in an external database (active) and files (passive) referenced from multi-media objects. A generic referencing mechanism is needed for classes whose instances will reference external objects, since the structure of references is dependent upon the external system.

remote objects:

For objects which will be referenced remotely, universal identifiers and UIDs need to be defined in the model. Since here we are talking about information which may move from one particular implementation environment to any other, both the UID and universal identifier need to be included in the model, to ensure standard understanding across sites.

The conclusions are as follows:

- references to both external and remote objects must be modelled explicitly;
- any object which will be potentially referenced remotely by another object, due to exchange or some other reason, must have a UID of some kind defined for it;
- any context containing GEHR objects must have an identifier of some kind defined for it (e.g. a server id and internet address).

Possibilities when moving a network of objects:

- external references (e.g. URIs) travel intact;
- internal references: break-reference (lose information) or convert to URI;
- to break a reference, convert it to reference a special “broken link” object which is understood by the architecture and can be manipulated in a standard way by applications;

A.5 Special Actions and Data

A.5.1 Term Sets

The model is designed with use of standard medical term sets in mind, and does not attempt directly to model anything from the medical or associated knowledge domains (e.g. surgery, physiology biology etc.). This specialist knowledge, which is expected to change and increase over time is simply used in a generic way by the model.

A.5.2 Querying

One of the basic uses for healthcare records is to satisfy queries about a patient or patients. The use of inheritance to model specialisation greatly facilitates this since entities which are a-kind-of another entity can be explicitly modelled. An example is the Transaction Cluster where various types of record information, familiar to practitioners are directly modelled. In an implementation of the model such as a database, efficient use of this knowledge can be made by applications, so as to find and process only specific types of Transaction.

The provision of various different clinical views of the information, such as a “nurses’ view” (limited to e.g. blood-pressure and temperatures) can be done very simply by the use of queries in intelligent applications, and it is specialisation in the OM which enables this.

A.5.3 External Information

There is certain information logically required in the EHCR which may exist outside the EHCR, and is known as “external information”. The structure and contents of such information are not in general known by the GOM, since they may be site-specific. Due to this, only a generic representation is possible in the GOM. The aim is to make such information exchangeable, just as for information completely modelled by the GOM (e.g. Transactions, HRIs), albeit at a much simpler semantic level.

Typical examples of such information include the following:

- Reference Data:
- HCP, HCF: only a very small number of attributes can be generalised for all GEHR sites, and while these can be modelled explicitly, other HCP data are likely to be completely site-specific and of no interest to other sites;
- Bulky Data:
- Multi-media: many multi-media objects will contain bulky data items whose contents cannot feasibly be contained in expanded form within the record;

There is a number of possibilities for where this kind of information is held, as follows:

- application and database implementations extend the GOM of such classes as the HCP to complete the model of HCP: no external data exist;
- in an external database (typically relational) accessible from the system in which GEHR applications exist;
- in an intelligent purpose-designed file system, such as the bulk-data systems for multi-media images etc.

The two exchange problems presented by this kind of information are as follows:

- since its format is opaque, how to exchange it;
- how to exchange other objects containing references to external objects.

A.5.4 Reference Data

The approach to modelling reference data is based on a number of considerations. Firstly, the requirement that at least some minimal contents of an exchanged reference object exist. For example, if the exchange of HCP objects did not imply the exchange of at least some standard minimum of information, e.g. name and position, target sites would effectively receive no information whatsoever, except perhaps for the external references from the source site. However, for these to be useful, each site would require knowledge of the schemas of external database systems at other sites, otherwise there would be no way of understanding the result.

The approach is as follows:

- reference data types are modelled in the OM with minimal classes, capturing the absolute minimum of semantics required at all sites. As with any class in the model, they can be extended for implementation at each site by the addition of new features;
- one of the attributes of these classes is a reference id for the external object. This provides the access mechanism to the rest of the external data, if any exist. In cases where there is none, the reference is simply empty. The type of this reference is likely to be STRING or URI (which covers the use of queries as identification), or some other basic generic type, since nothing can be assumed about the external system.
Note that in the case of queries as identification, the referenced “object” might be anything which could result from a query to the external system, from a single line of text, to a large report, to multi-media objects.
- another class feature is a function which when evaluated outputs the contents of the other class attributes (e.g. HCP.name, HCP.position etc.) in a standard form. This feature can also be used to access the external data via the reference id attribute, and format the information obtained in a standard way. Standardisation of the interface of such a feature is done by including its interface in the OM (as with any other feature) but leaving its specific implementation to site-specific applications.

Advantages:

- the limits of the OM are well-defined
- there is a guarantee of minimum contents, as with all other parts of model
- there is a guarantee of standard generic format of external information
- internal references do not change

A.5.5 Bulky Data

References to bulky data are handled simply by directly embedding them in the corresponding logical objects. Reference classes for bulky data within the OM are not necessary, because it is always assumed that the data are completely opaque: the only thing required to be known is the data format (e.g. JPEG, GIF etc.). Using this information, a receiver of such information can interpret bulky data, assuming it has tools capable of processing the specified format.

A.5.6 Multi Media

Multi-media information is seen by the GEHR model in a completely opaque fashion. Only a few generic details for multi-media items are stored; the rest is left up to applications. One multi-media format of special significance to the medical record is the medical drawing. This is often associated with applications which generate healthcare information using the drawing as the main user input mechanism. This could generate the same kind of data that other text- or menu-oriented interfaces might generate. The GOM makes no special provision for this, since it is assumed that where such information can be created in GOM-compliant format, it will be included in the record as if it was generated by any other means. The drawing itself (and any associated information mapping the GOM-compliant information to positions, colours etc. on the drawing), is simply stored in a nearby place in the record, and all intelligent processing is left to applications. The same reasoning can in fact be generalised to any multi-media type which might be used in interactive input.

A.5.7 Clinical Test Results

Test results will usually be stored in Report Transactions. They may be in any format, and may be converted to GOM-compliant format if need be. The model makes no special allowance for this: such conversions are left to applications.

A.5.8 Comment

It must be remembered that there may be many ways of expressing the same information. GEHR does not impose a particular method in any given instance but provides sufficient flexibility via the structures it offers to enable any HCP to arrange entries however he/she thinks appropriate.

GEHR thus embraces an architecture which can be used to define the progressive adherence to standards for the clinical content of records and for compliance with ethico-legal practice.

B A Worked Example of the Application of the GEHR Architecture to a Clinical Situation.

B.1 Scenario

Peter Grant registers with Dr. Kalra (The Heron Practice, London) on 24.10.94, the receptionist, Ms Smith, takes details of his name and address and he then sees the practice nurse for a health check.

The Nurse, Miss Judith Duddell, takes a history, checks for allergies and current medication, and carries out a physical examination. The patient then notices that his address is recorded wrongly. The nurse amends this mistake and then records the findings of her examination. She decides to record details of the patient's past medical history as summary data.

Mr. Grant comes in the next week to see Dr. Kalra about abdominal pain. He remembers his father used to get such pains and had a duodenal ulcer (which he forgot to tell the nurse during the health check). Dr. Kalra adds the new information linking it to the nurse's recording, makes an examination of Mr. Grant, diagnoses Acute Peptic Ulcer, and prescribes Cimetidine.

Later that year, Dr. Kalra receives a request from the Casualty department of the local hospital to send any records of Mr. Grant's having been given treatment for abdominal pain. Dr. Kalra extracts the relevant data from his records for sending to the hospital.

B.2 Outline of the Structures Used

This is an Outline of the Structures used in this example.

Only the main names and values of structures are shown to give the reader an overall picture of the worked example. A much fuller presentation of all the attributes of the structures used is given in the next Section.

The use of free text is denoted by the use of quotation marks: "free text abcd"

The use of terms from a termset is denoted by <term>.

HCF: (Heron Practice)

HCP: (Dr. Kalra)

HCP: (Nurse Duddell)

Staff_Member: (Mary Smith)

Patient: (Peter Grant)

EHCR_Source: (Heron Practice)

EHCR: (of Peter Grant)

Admin Transaction: (name, dob, gender from PATIENT)

HRI: <address1> "22,Perth Road"

HRI: <address2> "London"

HRI: <postcode> "N10 3DB"

Amended Admin Transaction:

HRI: <address1> "220 Perth Road"

Contact Transaction: (with Nurse Duddell)

Heading: <Medication>

HRI: <medication,current> <none>

Heading: "Test results"

Collection: <urinalysis>

HRI: <protein> <negative>

HRI: <glucose> <negative>

HRI: <bile> <not tested>

Collection: <Family>

HRI: <illness,serious> <none>

Heading: <Physical Examination>

HRI: <weight> 76 Kg

HRI: <height> 1.8 m

Collection: <blood pressure>

HRI: <systolic> 105 mmHg

HRI: <diastolic> 60 mm Hg

Summary Transaction

Heading: <History>

HRI: <allergies> <none>

HRI: <operations,previous> <none>

HRI: <illness,serious> <none>

Contact Transaction

(with Dr. Kalra)

Heading: <History>

Collection: <father>

HRI: <diagnosis> <duodenal ulcer>

*linked back to negative family history
recorded in the previous transaction*

Heading: <Symptoms>

Collection: <pain>

cx_comment: "worse at night"

HRI: <location> <epigastrium>

HRI: <duration> 4 days

Collection: <vomiting>

HRI: <last occurrence> - 4 days

HRI: <frequency> 2/day

HRI: <fever> <none>

HRI: <haematemesis> <none>

HRI: <comment> "I am worried"

info_provider <patient>

Heading: <Examination>

Collection: <tenderness>

HRI: <location> <epigastrium>

HRI: <severity> <severe>

Heading: <Diagnosis>

HRI: <Diagnosis> <peptic ulcer,acute>

Heading: <Treatment>

Collection: <prescription>

HRI: <drug name> <Cimetidine>

HRI: <dosage> 400mg/24hours

HRI: <instructions> <nocte>

HRI: <duration> 28 days

Report Transaction:

Collection: <request>

HRI: <requestor> “St. Bartholomew’s Casualty Department”

HRI: <date> 30.12.94

HRI <telephone message>“Please send any records of abdominal pain being treated in the last year”

EHCR_Extract:

Admin Transaction

Contact with Dr Kalra

B.3 Detailed description of Classes and Attributes

Each class in the example is given an internal Object Identifier (OID). An implementation would probably use such a means of recording the structure of the data but it is not part of the model (see Section Object Referencing, Appendix A.4). The OIDs are included to make the structure clearer to the reader. Attributes have been omitted if their value is void.

The use of free text is denoted by the use of quotation marks: "free text abcd"

The use of terms from a termset is denoted by <term>.

Square brackets are used to denote multi_text data types.

Health Care Facility (HCF) and "persons" (HCP, STAFF_MEMBER, PATIENT)

HCF OID=HCF-0001

Attributes

name: "The Heron Practice"
address: ADDRESS-0001

ADDRESS OID=ADDRESS-0001

Attributes

addr_lines: <street>, "101, Berry St", <town>, "London"
postcode: "EC4 9BD"
valid_from: 1.01.85

HCP OID=HCP-0001

Attributes

name: PERSON_NAME-0001
contact_nrs: [<home>, "123-4567"]
grade: "A1"
position: "Senior partner"
profession: <General Practitioner>
reg_country: <UK>
reg_number: GMC132435

PERSON_NAME OID=PERSON_NAME-0001

Attributes

surname: "Kalra"
forenames: ["Dipak"]
title: "Dr"

HCP OID=HCP-0002

Attributes

name: PERSON_NAME-0002
contact_nrs: [<home>, "765-4321"]
grade: "G"
position: "Practice Nurse"
profession: <State Registered Nurse>
reg_country: <UK>
reg_number: UKCC015432

PERSON_NAME OID=PERSON_NAME-0002

Attributes

surname: "Duddell"
 forenames: ["Judith","Ann"]
 title: "Miss"

STAFF_MEMBER OID=STAFF_MEMBER-0001

Attributes

name: PERSON_NAME-0003
 position: "Receptionist"

PERSON_NAME OID=PERSON_NAME-0003

Attributes

surname: "Smith"
 forenames: ["Mary","Jane"]
 title: "Ms"

PATIENT OID=PATIENT-0001

Attributes

name: PERSON_NAME-0004
 date_of_birth: 18.07.59
 gender: <male>

PERSON_NAME OID=PERSON_NAME-0004

Attributes

surname: "Grant"
 forenames: ["Peter"]
 title: "Mr"

The EHCR Source information

EHCR_Source OID=EHCR_Source-0001

Attributes

name: "The Heron Practice"
 location: ADDRESS-0001
 net_addrs: [heronp@server1.co.uk]
 hcps: [HCP-0001, HCP-0002]
 owning_hcf: HCF-0001
 ehcrs: [EHCR-0001]

The ECHR information.

There would usually be many EHCRs for any given EHCR_SOURCE. For the sake of the example, only one is given.

The EHCR contains five Versioned Transactions:-

1. The ADMIN created by the receptionist and later amended by the nurse.
2. The CONTACT created by the nurse.
3. The SUMMARY created by the nurse.
4. The CONTACT created by the doctor.
5. The REPORT created by the doctor for sending to the hospital.

EHCR OID=EHCR-0001

Attributes

ehcr_id: GEHR_EHCR-0001
dt_creation: 24.10.1994
hcp_created_by: HCP-0001
transactions: [VT-0001,VT-0002,VT-0003,VT-0004,VT-0005]

ADMIN - The first Versioned Transaction plus its associated Health Record Items (HRI). There is a single ADMIN Transaction which has been amended once to correct the address information. The original version was recorded by the receptionist. The amendment was made by the nurse when the mistake in the original was pointed out.

Versioned_Transaction OID=VT-0001

Attributes

uid: GEHR_VT-0001
dt_created: 24.10.1994 10:30
access_rights: <All Clinicians>,<Patient>
amend_rights: <All Clinicians>
gehr_version: 1.5
versions: ADMIN-0001

ADMIN OID=ADMIN-0001

Attributes

revision_id: 1.0
dt_committed: 24.10.1994 10:30
hcp_authorizing: HCP-0001
hcp_legally_resp: HCP-0001
type_of_change: <NEW>
recorder: STAFF_MEMBER-0001
subject: PATIENT-0001
items: [HRI-0001,HRI-0002,HRI-0003]

HRI OID=HRI-0001

Attributes

uid: GEHR_HRI-0001
name: <Address1>
emphasis: <low>
recorder: STAFF_MEMBER-0001
access_rights: <All Clinicians>,<Patient>
dt_observed: 24.10.94 10:30
certainty: <certain>
is_derived: FALSE
content: "22 Perth Road"

HRI OID=HRI-0002

Attributes

uid: GEHR_HRI-0002
name: <Address2>
emphasis: <low>
recorder: STAFF_MEMBER-0001
access_rights: <All Clinicians>,<Patient>
dt_observed: 24.10.94 10:30
certainty: <certain>
is_derived: FALSE
content: "London"

HRI OID=HRI-0003

Attributes

uid: GEHR_HRI-0003
name: <Postcode>
emphasis: <low>
recorder: STAFF_MEMBER-0001
access_rights: <All Clinicians>,<Patient>
dt_observed: 24.10.94 10:30
certainty: <certain>
is_derived: FALSE
content: "N10 3DB"

This is the amended ADMIN Transaction. It contains all the HRIs contained in the original version except for HRI-0001 (Address1) which was wrong. This has been replaced by HRI-0004 which contains the corrected information. Note that this method of modeling version control says nothing at all about implementation or the need for compression. See Section 5.3.1 (version control) for a fuller discussion.

ADMIN-0001 which was version 1.0 already existed. This entry now updates the number of versions to two.

ADMIN OID=ADMIN-0001

Attributes

revision_id: 1.1
dt_committed: 24.10.1994 10:55
hcp_authorizing: HCP-0002
hcp_legally_resp: HCP-0001
type_of_change: <amendment>
recorder: HCP-0002
subject: PATIENT-0001
items: [HRI-0002,HRI-0003,HRI-0004]

HRI OID=HRI-0004

Attributes

uid: GEHR_HRI-0004
name: <Address1>
emphasis: <low>
recorder: HCP-0002
access_rights: <All Clinicians>,<Patient>
dt_observed: 24.10.94 10:55
certainty: <certain>
is_derived: FALSE
content: "220 Perth Road"

CONTACT - The second Versioned Transaction plus its associated Health Record Items (HRI), Health Record Item Collections (HRI_COLLECTION) and Headings. There are two separate CONTACT Transactions in this EHCR. This is the one associated with the patient's consultation with the nurse. The other one (the fourth Versioned Transaction) is the one associated with the patient's later consultation with the doctor.

Versioned_Transaction OID=VT-0002

Attributes

uid: GEHR_VT-0002
dt_created: 24.10.1994 11:00
access_rights: <All Clinicians>,<Patient>
amend_rights: <All Clinicians>
gehr_version: 1.5
versions: CONTACT-0001

CONTACT OID=CONTACT-0001

revision_id: 1.0
 dt_committed: 24.10.1994 11:00
 hcp_authorizing: HCP-0002
 hcp_legally_resp: HCP-0001
 type_of_change: <NEW>
 recorder: HCP-0002
 dt_occurred: 24.10.94 10:33
 items: [HRI-0009,HRI_COLLECTION-0001, HRI_COLLECTION-0002,
 HRI-0014,HRI-0015, HRI_COLLECTION-0003]

HEADING OID=HEADING-0001

Attributes

uid: GEHR_HEADING-0001
 name: <medication>
 emphasis: <low>
 recorder: HCP-0002

HRI OID=HRI-0009

Attributes

uid: GEHR_HRI-0009
 name: <Medication(current)>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [H-0001]
 dt_observed: 24.10.94 10:45
 certainty: <certain>
 is_derived: FALSE
 content: <none>

HEADING OID=HEADING-0002

Attributes

uid: GEHR_HEADING-0002
 name: <Test results>
 emphasis: <low>
 recorder: HCP-0002

HRI_COLLECTION OID=HRI_COLLECTION-0001

Attributes

uid: GEHR_HRI_COLLECTION-0001
 name: <Urinalysis>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [H-0002]
 members: [HRI-0010, HRI-0011, HRI-0012]

HRI OID=HRI-0010

Attributes

uid: GEHR_HRI-0010
Name: <Protein>
emphasis: <low>
recorder: HCP-0002
access_rights: <All Clinicians>,<Patient>
dt_observed: 24.10.94 10:45
certainty: <ertain>
is_derived: FALSE
content: <negative>

HRI OID=HRI-0011

Attributes

uid: GEHR_HRI-0011
name: <Glucose>
emphasis: <low>
recorder: HCP-0002
access_rights: <All Clinicians>,<Patient>
dt_observed: 24.10.94 10:45
certainty: <ertain>
is_derived: FALSE
content: <negative>

HRI OID=HRI-0012

Attributes

uid: GEHR_HRI-0012
name: <bile>
emphasis: <low>
recorder: HCP-0002
access_rights: <All Clinicians>,<Patient>
dt_observed: 24.10.94 10:45
certainty: <ertain>
is_derived: FALSE
content: <not tested>

HRI_COLLECTION OID=HRI_COLLECTION-0002

Attributes

uid: GEHR_HRI_COLLECTION-0002
name: <Family>
emphasis: <low>
recorder: HCP-0002
access_rights: <All Clinicians>,<Patient>
members: [HRI-0013]

HRI OID=HRI-0013

Attributes

uid: GEHR_HRI-0013
 name: <illness(serious)>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 24.10.94 10:45
 certainty: <ertain>
 is_derived: FALSE
 content: <none>

HEADING OID=HEADING-0003

Attributes

uid: GEHR_HEADING-0003
 name: <physical examination>
 emphasis: <low>
 recorder: HCP-0002

HRI OID=HRI-0014

Attributes

uid: GEHR_HRI-0014
 name: <weight>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [H-003]
 ct_comment: "with light clothes"
 dt_observed: 24.10.94 10:45
 certainty: <ertain>
 is_derived: FALSE
 content: 76kg {Quantity - detail of modeling omitted}

HRI OID=HRI-0015

Attributes

uid: GEHR_HRI-0015
 name: <height>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [H-003]
 dt_observed: 24.10.94 10:45
 certainty: <ertain>
 is_derived: FALSE
 content: 1.80m {Quantity - detail of modeling omitted}

HRI_COLLECTION OID=HRI_COLLECTION-0003

Attributes

uid: GEHR_HRI_COLLECTION-0003
 name: <Blood pressure>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [H-003]
 members: [HRI-0016, HRI-0017]

HRI OID=HRI-0016

Attributes

uid: GEHR_HRI-0016
 name: <blood pressure (systolic)>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 24.10.94 10:45
 certainty: <certain>
 is_derived: FALSE
 content: 105mmHg {Quantity - detail of modeling omitted}

HRI OID=HRI-0017

Attributes

uid: GEHR_HRI-0017
 Name: <blood pressure (diastolic)>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 con_comment: "difficult to hear"
 dt_observed: 24.10.94 10:45
 certainty: <uncertain>
 is_derived: FALSE
 content: 60mmHg {Quantity - detail of modeling omitted}

SUMMARY - The third Versioned Transaction plus its associated Health Record Items (HRI) and Headings. There is a single SUMMARY Transaction in this EHCR. This is the one associated with the patient's consultation with the nurse. She made the decision to record past medical history as SUMMARY information. Had she found evidence of allergies, she may well have decided to record this as NOTA BENE information.

Versioned_Transaction OID=VT-0003

Attributes

uid: GEHR_VT-0003
 dt_created: 24.10.94 11:15
 access_rights <All Clinicians>,<Patient>
 amend_rights: <All Clinicians>
 gehr_version: 1.5
 versions: SUMMARY-0001

SUMMARY OID=SUMMARY-0001

revision_id: 1.0
 dt_committed: 24.10.94 11:15
 hcp_authorizing: HCP-0002
 hcp_legally_resp: HCP-0001
 type_of_change: <NEW>
 recorder: HCP-0002
 items: [HRI-0018, HRI-0019, HRI-0020]

HEADING OID=HEADING-0004

Attributes

uid: GEHR_HEADING-0004
 name: <history>
 emphasis: <low>
 recorder: HCP-0002

HRI OID=HRI-0018

Attributes

uid: GEHR_HRI-0018
 name: <allergies>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HEADING-0004]
 dt_observed: 24.10.94 11:15
 certainty: <ertain>
 is_derived: FALSE
 content: <none>

HRI OID=HRI-0019

Attributes

uid: GEHR_HRI-0019
 name: <operations (previous)>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HEADING-0004]
 dt_observed: 24.10.94 11:15
 certainty: <ertain>
 is_derived: FALSE
 content: <none>

HRI OID=HRI-0020

Attributes

uid: GEHR_HRI-0020
 name: <illness (serious)>
 emphasis: <low>
 recorder: HCP-0002
 access_rights: <All Clinicians>,<Patient>

annotated_by: [HEADING-0004]
 dt_observed: 24.10.94 11:15
 certainty: <ertain>
 is_derived: FALSE
 content: <none>

CONTACT - The fourth Versioned Transaction plus its associated Health Record Items (HRI), Health Record Item Collections (HRI_COLLECTION) and Headings. This is the CONTACT associated with the patient's consultation with the doctor.

Versioned_Transaction OID=VT-0004

Attributes

uid: GEHR_VT-0004
 dt_created: 31.10.94 9:30
 access_rights <All Clinicians>,<Patient>
 amend_rights: <All Clinicians>
 gehr_version: 1.5
 versions: CONTACT-0002

CONTACT OID=CONTACT-0002

revision_id: 1.0
 dt_committed: 31.10.94 9:30
 hcp_authorizing: HCP-0001
 hcp_legally_resp: HCP-0001
 type_of_change: <NEW>
 recorder: HCP-0001
 dt_occurred: 31.10.94 9:15
 items: [HRI_COLLECTION-0004, HRI_COLLECTION-0005,
 HRI_COLLECTION-0006, HRI-0026, HRI-0027,HRI-0028,
 HRI_COLLECTION-0007, HRI-0031,HRI_COLLECTION-0008]

HEADING OID=HEADING-0005

Attributes

uid: GEHR_HEADING-0005
 name: <history>
 emphasis: <low>
 recorder: HCP-0001

HRI_COLLECTION OID=HRI_COLLECTION-0004

Attributes

uid: GEHR_HRI_COLLECTION-0004
 name: <father>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HEADING-0005]
 members: [HRI-0021]

HRI OID=HRI-0021

Attributes

uid: GEHR_HRI-0021
 name: <diagnosis>
 emphasis: <low>
 recorder: HCP-0001
 links: [HRI-0013]
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 31.10.94 9:30
 certainty: <ertain>
 is_derived: FALSE
 content: <duodenal ulcer>

HEADING OID=HEADING-0006

Attributes

uid: GEHR_HEADING-0006
 name: <symptoms>
 emphasis: <low>
 recorder: HCP-0001

HRI_COLLECTION OID=HRI_COLLECTION-0005

Attributes

uid: GEHR_HRI_COLLECTION-0005
 name: <pain>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HREADING-0006]
 cx_comment: "worse at night"
 members: [HRI-0022, HRI-0023]

HRI OID=HRI-0022

Attributes

uid: GEHR_HRI-0022
 name: <location>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 31.10.94 9:30
 certainty: <ertain>
 is_derived: FALSE
 content: <epigastrium>

HRI OID=HRI-0023

Attributes

uid: GEHR_HRI-0023
 name: <duration>
 emphasis: <low>
 recorder: HCP-0001

access_rights: <All Clinicians>,<Patient>
 dt_observed: 31.10.94 9:30
 certainty: <certain>
 is_derived: FALSE
 content: 4 days {Quantity - detail of modeling omitted}

HRI_COLLECTION OID=HRI_COLLECTION-0006

Attributes

uid: GEHR_HRI_COLLECTION-0006
 name: <vomiting>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HEADING-0006]
 members: [HRI-0024, HRI-0025]

HRI OID=HRI-0024

Attributes

uid: GEHR_HRI-0024
 name: <last occurrence>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 31.10.94 9:30
 certainty: <certain>
 is_derived: FALSE
 content: -4 days {Quantity - detail of modeling omitted}

HRI OID=HRI-0025

Attributes

uid: GEHR_HRI-0025
 name: <last occurrence>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 31.10.94 9:30
 certainty: <certain>
 is_derived: FALSE
 content: 2/day {Quantity - detail of modeling omitted}

HRI OID=HRI-0026

Attributes

uid: GEHR_HRI-0026
 name: <fever>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HEADING-0006]

dt_observed: 31.10.94 9:30
 certainty: <ertain>
 is_derived: FALSE
 content: <none>

HRI OID=HRI-0027

Attributes

uid: GEHR_HRI-0027
 name: <haematemesis>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HEADING-0006]
 dt_observed: 31.10.94 9:30
 certainty: <ertain>
 is_derived: FALSE
 content: <none>

HRI OID=HRI-0028

Attributes

uid: GEHR_HRI-0028
 name: <comment>
 emphasis: <low>
 recorder: HCP-0001
 info_prov: <PERSON_NAME-0004>
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HEADING-0006]
 dt_observed: 31.10.94 9:30
 certainty: <ertain>
 is_derived: FALSE
 content: "I am worried"

HEADING OID=HEADING-0007

Attributes

uid: GEHR_HEADING-0007
 name: <examination>
 emphasis: <low>
 recorder: HCP-0001

HRI_COLLECTION OID=HRI_COLLECTION-0007

Attributes

uid: GEHR_HRI_COLLECTION-0007
 name: <Tenderness>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HEADING-0007]
 members: [HRI-0029,HRI-0030]

HRI OID=HRI-0029

Attributes

uid: GEHR_HRI-0029
name: <location>
emphasis: <low>
recorder: HCP-0001
access_rights: <All Clinicians>,<Patient>
dt_observed: 31.10.94 9:30
certainty: <certain>
is_derived: FALSE
content: <epigastrium>

HRI OID=HRI-0030

Attributes

uid: GEHR_HRI-0030
name: <severity>
emphasis: <low>
recorder: HCP-0001
access_rights: <All Clinicians>,<Patient>
dt_observed: 31.10.94 9:30
certainty: <certain>
is_derived: FALSE
content: <severe>

HEADING OID=HEADING-0008

Attributes

uid: GEHR_HEADING-0008
name: <diagnosis>
emphasis: <low>
recorder: HCP-0001

HRI OID=HRI-0031

Attributes

uid: GEHR_HRI-0031
name: <Diagnosis>
emphasis: <low>
recorder: HCP-0001
access_rights: <All Clinicians>,<Patient>
annotated_by: [HEADING-0008]
dt_observed: 31.10.94 9:30
certainty: <certain>
is_derived: FALSE
content: <peptic ulcer,acute>

HEADING OID=HEADING-0009

Attributes

uid: GEHR_HEADING-0009
 name: <treatment>
 emphasis: <low>
 recorder: HCP-0001

HRI_COLLECTION OID=HRI_COLLECTION-0008

Attributes

uid: GEHR_HRI_COLLECTION-0008
 name: <Prescription>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 annotated_by: [HEADING-0009]
 members: [HRI-0032, HRI-0033, HRI-0034, HRI-0035]

HRI OID=HRI-0032

Attributes

uid: GEHR_HRI-0032
 name: <drug name>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 31.10.94 9:30
 certainty: <certain>
 is_derived: FALSE
 content: <cimetidine>

HRI OID=HRI-0033

Attributes

uid: GEHR_HRI-0033
 name: <dosage>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 31.10.94 9:30
 certainty: <certain>
 is_derived: FALSE
 content: 400mg/24 hrs {Quantity - detail of modeling omitted}

HRI OID=HRI-0034

Attributes

uid: GEHR_HRI-0034
 name: <frequency>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 31.10.94 9:30

certainty: <ertain>
 is_derived: FALSE
 content: <nocte>

HRI OID=HRI-0035

Attributes

uid: GEHR_HRI-0035
 name: <duration>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 dt_observed: 31.10.94 9:30
 certainty: <ertain>
 is_derived: FALSE
 content: 28 days {Quantity - detail of modeling omitted}

REPORT - The fifth Versioned Transaction records the result of a telephone message from the hospital, taken by the receptionist. It is an example of a Report Transaction.

Versioned_Transaction OID=VT-0005

Attributes

uid: GEHR_VT-0005
 dt_created: 30.12.1994 13:30
 access_rights <All Clinicians>,<Patient>
 amend_rights: <All Clinicians>
 gehr_version: 1.5
 versions: REPORT-0001

REPORT OID=REPORT-0001

Attributes

revision_id: 1.0
 dt_committed: 30.12.1994 13:30
 hcp_authorizing: HCP-0001
 hcp_legally_resp: HCP-0001
 type_of_change: <NEW>
 recorder STAFF_MEMBER-0001
 in_reply_to: HRI-0036
 items: [HRI_COLLECTION-0009]

HRI_COLLECTION OID=HRI_COLLECTION-0009

Attributes

uid: GEHR_HRI_COLLECTION-0009
 name: <Request>
 emphasis: <low>
 recorder: HCP-0001
 access_rights: <All Clinicians>,<Patient>
 members: [HRI-0036, HRI-0037, HRI-0038]

HRI OID=HRI-0036

Attributes

uid: GEHR_HRI-0036
name: <requestor>
emphasis: <low>
recorder: HCP-0001
access_rights: <All Clinicians>,<Patient>
dt_observed: 30.12.94 13:30
certainty: <ertain>
is_derived: FALSE
content: "St. Bartholomew's Casualty Department"

HRI OID=HRI-0037

Attributes

uid: GEHR_HRI-0037
name: <date>
emphasis: <low>
recorder: HCP-0001
access_rights: <All Clinicians>,<Patient>
dt_observed: 30.12.94 13:30
certainty: <ertain>
is_derived: FALSE
content: 30.12.94

HRI OID=HRI-0038

Attributes

uid: GEHR_HRI-0038
name: <telephone message>
emphasis: <low>
recorder: HCP-0001
access_rights: <All Clinicians>,<Patient>
dt_observed: 30.12.94 13:30
certainty: <ertain>
is_derived: FALSE
content: "Please send any records of abdominal pain being treated in the last year"

EHCR_EXTRACT - This is the data associated with responding to the request from the hospital's Casualty Department. The doctor selects the relevant Transactions and puts them into an EHCR_EXTRACT. For identification, the current ADMIN Transaction is included. Note that any terms in any of the data sent may be sent in expanded form, coded form or a combination of both. All associated HRIs, HRI_COLLECTIONS and PERSON objects will also be sent.

EHCR_EXTRACT

Attributes

ehcr_id:	GEHR_EHCR_EXTRACT-0001
dt_creation:	30.12.1994
hcp_created_by:	HCP-0001
people:	[HCP-0001,HCP-0002,STAFF_MEMBER-0001]
hcfs:	[HCF-0001]
transactions:	[VT-0001,VT-0004]

C Other Considerations

C.1 Stages in the Evolution of the GEHR Architecture

The Interim GEHR Architecture was expressed in largely descriptive clinical terms. Although the IGA met the objectives of providing sufficient flexibility to accommodate all the potential individual styles of recording, and defining a set of constructs with which to model the data and concepts used in clinical practice, there were weaknesses in the way ideas were expressed and communicated.

During the project, the proposed architectural constructs have gradually moved from a clinical to an informatics authorship, and have been subjected to the application of formal modelling methodologies. The University of Hull Medical Information Engineering Group, responsible for overseeing quality assurance within the project, produced a set of development guidelines to encourage the use of appropriate techniques and methodologies, and to ensure adherence to appropriate standards (e.g. ISO 9000-3, etc.).

The requirement to interface and communicate with other systems and to achieve compliance with existing and emerging standards and protocols in areas such as communication, image handling (as well as international safety and quality standards) demands a formal and rigorous architecture definition.

A number of participants have examined ways of interfacing the emerging GEHR architecture with legacy systems. It is acknowledged that there is considerable investment in existing systems: they represent a valuable resource. It is unlikely that people will consider transferring to a new one unless a strong justification can be made on technical as well as clinical and cost-benefit grounds. For further details, see Chapter 9. The culmination of all these efforts is also presented in GEHR Deliverable 18 [12].

GEHR has maintained close links with a number of other European projects, for example GALEN, DILEMMA, GAMES, OPENLABS, TANIT, DIABCARD, NUCLEUS, EURIPACS, OEDIPE and SEISMED. Several members of the consortium are represented on Working Groups of CEN TC/251 (e.g. WGs 1, 3, 5, 6, and 7). To communicate ideas in such contexts in an effective, concise and unambiguous way, formal expressions, particularly graphical ones, have much to offer.

The trend towards the adoption of formal methods has been fundamental to the emergence of the final GOM (GEHR Object Model) presented in Chapter 6 of this document. There has been a gradual evolution in thinking from the CHRA (Current Health Record Architecture used in Health.one) data structure through successive generations of IGA (Interim GEHR Architecture) to the final GOM. This evolution is chronicled in GEHR Deliverable 15 [13].

The GEHR project has tackled an ambitious objective of providing a general means of handling a wide variety of clinical information from disparate clinical fields and a multitude of systems currently in use. In seeking to achieve a generality in application, it has faced a dilemma in the expression of its conclusions and proposals. The choice of an appropriate information model has been fundamental to this. The primary use of a model is to capture the thinking about a problem in

such a way that another person can fully understand that problem and any solution proffered.

It was recognised that modelling techniques are needed to allow the expression of the range of information structures which the project functionality demands while providing the transferability of meaning required. Such techniques would allow for:

- abstraction (separating implementation concerns from content definition on the GEHR record);
- formality (for example specifying the structure of the record in a way capable of describing all possible records, rather than by example);
- an object oriented approach.

The end objective is a formal model of the data defined in terms of object classes and structures which capture the full semantic richness of the clinical and ethico-legal requirements. This is likely to be an important requirement for a formalism to anchor the future common health record architecture for Europe with capability for monitoring of compliance to specified standards. It should improve the communication of ideas and also increase the acceptability of the architecture as far as the wider computing and healthcare communities are concerned.

One particular application of this model is in deriving a formal view which can constitute an exchange format whereby safe and rigorous exchange of clinical data may be undertaken.

To assist the reader to a better understanding of the resultant GOM, Chapter 6 introduces the particular formalism chosen to represent the model - its evolution, its features, properties and capabilities, and a brief explanation of its syntax. Appendix A covers formal modelling generally and summarises the benefits of formal modelling in general and object modelling in particular.

The path towards this Deliverable has obviously not been straightforward. There are many alternative approaches to almost every issue. The primary purpose of this document has been to show the solutions to which the work has led and NOT the route taken, with its many dead ends and wrong turns.

This document is aimed at technical readers who want something concise and useful to work with and to implement. It would be inappropriate to incorporate all the different alternatives and mistaken paths that have been followed *en route*. Nonetheless, it is important that people reading this document are able to find all the reasons behind particular choices and solutions. Further detail is in a separate document currently in progress and many of the issues are explored further in [13], but a number of examples are given below to illustrate how it has been done.

The danger in presenting only the agreed solution is that it may appear that obvious paths have been forgotten. This has not been the case. The “obvious” solutions have been explored and thought through in detail. Where they have been rejected, it has been for a good reason - that they do not perhaps cover all eventualities in the medical arena. These issues divide into six general areas:

C.2 Future Work

A three year project cannot cover all aspects in detail and some have deliberately been left for completion for a follow-on project and a demonstrator phase.

Examples include:

- formalism (see also The Way Forward - Chapter 10);
- functions: an object model of this nature would usually contain many more functions; as there was insufficient resource to investigate these in full, it was decided to leave this aspect for the follow-on project.

C.3 Minimum and Compromise Solutions

Where appropriate, a minimal solution has been chosen over a more complex one because:

- it will be more easily formalised;
- it will be easier to modify if needed;

Examples include:

Triggers:

The full richness of triggers requires much work on the best way to store the conditions and actions required. Rather than suggest that a partial solution is available, without having studied the issue in depth, GEHR has provided a Transaction type for this purpose which will be expanded in future work.

HCP Class:

In order to allow more subdivisions of access permissions to be shared, it is necessary to subdivide the HCP class - depending on the type of clinician. The best way of doing this is not yet known, so the safest solution at this time is to provide access (at the shared level) to the authorising HCP (optionally plus recorder, HCP responsible for care, patient...) or everyone.

Patient:

The model of Patient may be extended to include many features that might be seen as important. However, the model only includes those features without which it is known that no safe identification of the patient could be made. This class could be extended in the future or on a local system. All other details that might need to be recorded about the patient for identification follow the normal mechanism of Observations in an Administrative Transaction.

C.4 Results of Others' Work

It was inevitable that the work of GEHR would impinge on other areas and it has been seen as pointless to try to repeat the work of other projects, particularly where there was, in any case, insufficient resource. Wherever possible, place holders have been left to allow for the future incorporation of results of the work of other projects.

Examples:

- Termsets - (See also Evolution of TermSets);
- Patient ID - Many projects are working on this issue;
- Protocols - Other projects (e.g. DILEMMA) have addressed this.

C.5 Particular Solutions Chosen

Sometimes the project has been faced with alternatives where it has not been possible to say that one is clearly 'better' in any sense than another. In these cases, a particular solution has been chosen.

Examples:

- Arguments can be put for and against allowing the recording of original language at different levels e.g. an entire Observation or each individual piece of text within an Observation. The solution considered to be the most flexible was chosen at this time.
- Similar decisions were taken for emphasis and access rights

C.6 Ideas Rejected as Inadequate or Wrong

Some apparently obvious solutions were found to be inadequate or wrong.

Examples:

'content properties'

The possibility of allowing certain aspects of a named Observation to be attributes of that Observation was explored but the ramification was that every Item would have had to be a separate class.

The quantity cluster

Originally it was thought that this could be modelled by a value and a unit term. This is, however, totally inadequate (see Evolution of Term Sets - Chapter 8)

Building Elements, Building Element Groups and GEHR Logical Types

Initial attempts to model the architectural components and logical types resulted in attempts to model at an inappropriate level i.e. at a level in fact appropriate to an implementation or exchange format, or else so generic as to allow nonsensical recordings such as 12th Aug kg m⁻².

order, screen presentation

Originally it was thought that the order and style (e.g. colour, font) of Entries was a significant part of the underlying information. This is not the case. For a full discussion, see the actual GOM description.

the update concept,

It was thought at one stage, that the amendment of Transactions should be modelled in a variety of ways. However, it became apparent that the simple means proposed covered all eventualities.

Headings/Collections

Originally, no distinction was made between Headings and HRI Collections. However, in the process of testing the model with numerous examples, it became clear that the nature of Headings was distinct from that of medical Observations. These have now been modelled separately.

C.7 Outside the record

Some areas, outside the official record have nevertheless been studied in detail.

Examples:

- views, templates
- incomplete (not yet committed) Transactions
- shadow (student) notes

Future work of the GEHR consortium and others will develop and document this work.

D Glossary of Acronyms

ASN.1	Abstract Syntax Notation 1
CEN	Comité Européen de Normalisation
CHRA	Current Health Record Architecture
CORBA	Common Object Request Broker: Architecture & Spec.
EBNF	Extended Backus Naur Form
EC	European Commission
EDI	Electronic Data Interchange
EHCR	Electronic Health Care Record
EHCRA	EHCR Architecture (also name of the first EHCR Architecture standard from CEN TC251/WG1/PT011)
EHCREA	Proposed Extended EHCRA from CEN
EU	European Union
FTP	File Transfer Protocol
GEF	GEHR Exchange Format
GEHR	Good European Health Record
GOM	GEHR Object Model
GUI	Graphical User Interface
H	= HRI
HC	= HRI_COLLECTION
HCF	Health Care Facility
HCR	Health Care Record
HCP	Health Care Professional
HELIOS	Hospital Environment Language within an Information Object System
HL7	Health Level 7
HRI	Health Record Item
HRI_COLLECTION	HRI Collection
IGA	Interim GEHR Architecture
ISAM	Indexed Sequential Access Method
JPEG	Joint Photographic Expert Group
OM	Object Model
OO	Object Oriented
OOPL	Object Oriented Programming Language
OPADE	Optimization of Drug Prescriptions using Advanced Informatics - AIM A2027
OPENLABS	Application of Advanced Informatics and Telematics for Optimization of Clinical Laboratories - AIM A2028
RDBMS	Relational DataBase Management System
SCPECG	Standard Communications Protocol for Computerised Electro-Cardiography
UOI	Unique Object Identifier
URI	Universal Resource Identifier
URL	Uniform Resource Locators
WG	Working Group
WP	Word Processor
WWW	World Wide Web