



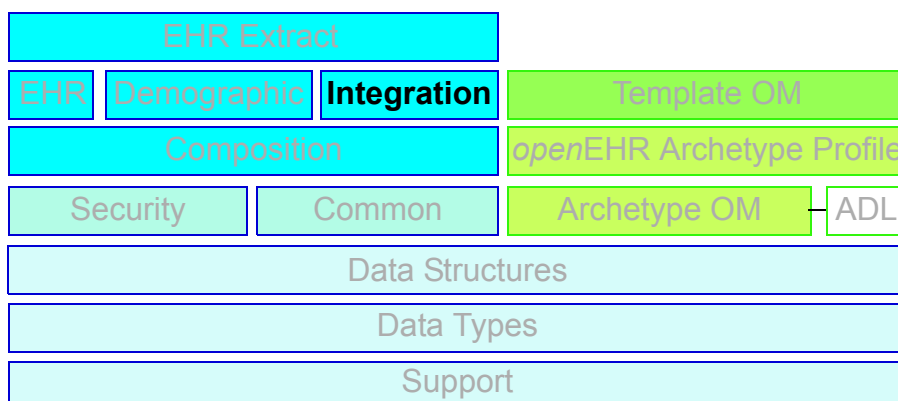
The *openEHR* Reference Model

# Integration Information Model

Editors: T Beale <sup>a</sup>		
Revision: 0.6	Pages: 15	Date of issue: 22 Jul 2006

a. Ocean Informatics

**Keywords:** EHR, reference model, integration, EN13606, openehr



© 2003-2007 The *openEHR* Foundation.

The *openEHR* Foundation is an independent, non-profit community, facilitating the sharing of health records by consumers and clinicians via open-source, standards-based implementations.

**Founding Chairman** David Ingram, Professor of Health Informatics, CHIME, University College London

**Founding Members** Dr P Schloeffel, Dr S Heard, Dr D Kalra, D Lloyd, T Beale

**email:** [info@openEHR.org](mailto:info@openEHR.org) **web:** <http://www.openEHR.org>

## Copyright Notice

© Copyright openEHR Foundation 2001 - 2007  
All Rights Reserved

1. This document is protected by copyright and/or database right throughout the world and is owned by the openEHR Foundation.
2. You may read and print the document for private, non-commercial use.
3. You may use this document (in whole or in part) for the purposes of making presentations and education, so long as such purposes are non-commercial and are designed to comment on, further the goals of, or inform third parties about, openEHR.
4. You must not alter, modify, add to or delete anything from the document you use (except as is permitted in paragraphs 2 and 3 above).
5. You shall, in any use of this document, include an acknowledgement in the form: "© Copyright openEHR Foundation 2001-2007. All rights reserved. [www.openEHR.org](http://www.openEHR.org)"
6. This document is being provided as a service to the academic community and on a non-commercial basis. Accordingly, to the fullest extent permitted under applicable law, the openEHR Foundation accepts no liability and offers no warranties in relation to the materials and documentation and their content.
7. If you wish to commercialise, license, sell, distribute, use or otherwise copy the materials and documents on this site other than as provided for in paragraphs 1 to 6 above, you must comply with the terms and conditions of the openEHR Free Commercial Use Licence, or enter into a separate written agreement with openEHR Foundation covering such activities. The terms and conditions of the openEHR Free Commercial Use Licence can be found at [http://www.openehr.org/free\\_commercial\\_use.htm](http://www.openehr.org/free_commercial_use.htm)

## Amendment Record

Issue	Details	Who	Completed
<b>R E L E A S E 1.0.1</b>			
0.6	<b>CR-000203:</b> Release 1.0 explanatory text improvements. Added section on <i>openEHR</i> Extract. Added integration architecture diagram	T Beale	22 Jul 2006
<b>R E L E A S E 1.0</b>			
0.5	Initial Writing	T Beale	15 Sep 2005
<b>R E L E A S E 0.96</b>			

## Acknowledgements

The work reported in this paper has been funded by The University College, London and Ocean Informatics, Australia.



## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>7</b>
1.1	Purpose.....	7
1.2	Related Documents .....	7
1.3	Status.....	7
1.4	Peer review .....	7
1.5	Conformance.....	7
<b>2</b>	<b>Integration Package .....</b>	<b>9</b>
2.1	Requirements .....	9
2.2	Design Basis .....	9
2.2.1	Overview.....	9
2.2.2	Semantics of GENERIC_ENTRY .....	10
2.2.3	Use with openEHR Extracts .....	11
2.2.4	Integration with CEN EN13606 .....	11
2.3	Data Conversion Architecture .....	11
2.4	Class Descriptions.....	12
2.4.1	GENERIC_ENTRY Class .....	12
<b>A</b>	<b>References .....</b>	<b>13</b>
A.1	Standards.....	13



# 1 Introduction

---

## 1.1 Purpose

This document describes the architecture of the *openEHR* Integration Information Model, designed for use in legacy and other integration situations.

The intended audience includes:

- Standards bodies producing health informatics standards;
- Software development groups using *openEHR*;
- Academic groups using *openEHR*;
- The open source healthcare community;
- Medical informaticians and clinicians interested in health information;
- Health data managers.

## 1.2 Related Documents

Prerequisite documents for reading this document include:

- The *openEHR* Architecture Overview
- The *openEHR* Reference Model documents

## 1.3 Status

This document is under development, and is published as a proposal for input to standards processes and implementation works.

This document is available at [http://svn.openehr.org/specification/TAGS/Release-1.0.1/publishing/architecture/rm/integration\\_im.pdf](http://svn.openehr.org/specification/TAGS/Release-1.0.1/publishing/architecture/rm/integration_im.pdf).

The latest version of this document can be found at [http://svn.openehr.org/specification/TRUNK/publishing/architecture/rm/integration\\_im.pdf](http://svn.openehr.org/specification/TRUNK/publishing/architecture/rm/integration_im.pdf).

## 1.4 Peer review

Areas where more analysis or explanation is required are indicated with “to be continued” paragraphs like the following:

To Be Continued:      more work required

Reviewers are encouraged to comment on and/or advise on these paragraphs as well as the main content. Please send requests for information to [info@openEHR.org](mailto:info@openEHR.org). Feedback should preferably be provided on the mailing list [openehr-technical@openehr.org](mailto:openehr-technical@openehr.org), or by private email.

## 1.5 Conformance

Conformance of a data or software artifact to an *openEHR* Reference Model specification is determined by a formal test of that artifact against the relevant *openEHR* Implementation Technology Specification(s) (ITSs), such as an IDL interface or an XML-schema. Since ITSs are formal, automated derivations from the Reference Model, ITS conformance indicates RM conformance.





## 2 Integration Package

---

### 2.1 Requirements

Getting data in and out of the EHR is one of the most basic requirements *openEHR* aims to satisfy. In “greenfield” (new build) situations, and for data being created by GUI applications via the *openEHR* EHR APIs, there is no issue, since native *openEHR* structures and semantics are being used. In almost all other situations, existing data sources and sinks have to be accounted for. In general, external or ‘legacy’ data (here the term is used for convenience, and does not imply anything about the age or quality of the systems in question) have different syntactic and semantic formats than *openEHR* data, and seamless conversion requires addressing both levels.

Typical examples of legacy data sources and sinks include relational databases, HL7v2 messages, and HL7 CDA documents. HL7v2 messages are probably one of the most common sources of pathology messages in many countries; EDIFACT messages are another. More recently, HL7v2 messages have been designed for referrals and even discharge summaries. Not all legacy systems are standardised; many if not most hospitals as well as GP and other desktop products have their own private models of data and terminology usage. Technically speaking, there is not much difference between standardised and non-standardised legacy models; only the reusability of the solution differs.

Another important category of externally sourced data addressed by the Integration package described here is data expressed in a form of a CEN EN13606 Extract. Part 1 of EN13606 defines an information model which is nearly identical to that of *openEHR* at the `COMPOSITION` and `SECTION` levels. The CEN EN13606 `Entry` class is a generic structure with a minimum of contextual meta-data, and can easily be mapped to the *openEHR* `Entry` type described in this specification.

The primary need with respect to legacy data is to be able to convert data from multiple mutually incompatible sources into a single, standardised patient-centric EHR for each patient, that can then be longitudinally viewed and queried. This is what enables GP and specialist notes, diagnoses and plans to be integrated with laboratory results from multiple sources, patient notes, administrative data and so on, to provide a coherent record of the patient journey.

In technical terms, a number of types of incompatibility have to be dealt with. There is no guarantee of correspondence of scope of incoming transactions and target *openEHR* structures - an incoming document for example might correspond to a number of clinical archetypes. Structure will not usually correspond, with legacy data (particularly messages) usually having flatter structures than those defined in target archetypes. Terminology use is extremely variable in existing systems and messages, and also has to be dealt with. Data types will also not correspond directly, so that for example, a mapping between an incoming string “110/80 mmHg” and the target *openEHR* form of two `DV_QUANTITY` objects each with their own value and units has to be made.

### 2.2 Design Basis

#### 2.2.1 Overview

The design basis for connecting existing systems to *openEHR* is founded upon a clear separation of the syntactic and semantic transformations required on data. The syntactic transformation converts source data from its original form (or whatever intermediate form it may have been converted to) to a format obeying a special class in the *openEHR* reference model, but whose logical structure and semantics are controlled by ‘integration’ archetypes so as to mimic the design of the source data. This step brings the data into the *openEHR* computational context. The second step causes transformation

on this intermediate *openEHR* data into data which are a) instances of the main *openEHR* reference model, and b) obey ‘designed’ clinical archetypes.

The additional elements of the *openEHR* architecture which make this transformation possible are:

- a class `GENERIC_ENTRY`, which is a sibling of `SECTION` and `ENTRY`, and contains completely generic, archetypable structures;
- ‘integration’ archetypes, i.e. archetypes defined against the `GENERIC_ENTRY` class;
- semantic transformation rules from *openEHR* data based on `GENERIC_ENTRY` and integration archetypes to data based on the subtypes of `ENTRY`, and designed archetypes.

FIGURE 1 illustrates the `rm.integration` package, which contains a single class `GENERIC_ENTRY`. Unlike other classes in the *openEHR* reference model, `GENERIC_ENTRY` contains no hard-wired attributes at all, only one generic attribute, *data*. No assumptions at all are made about the actual shape of such data.

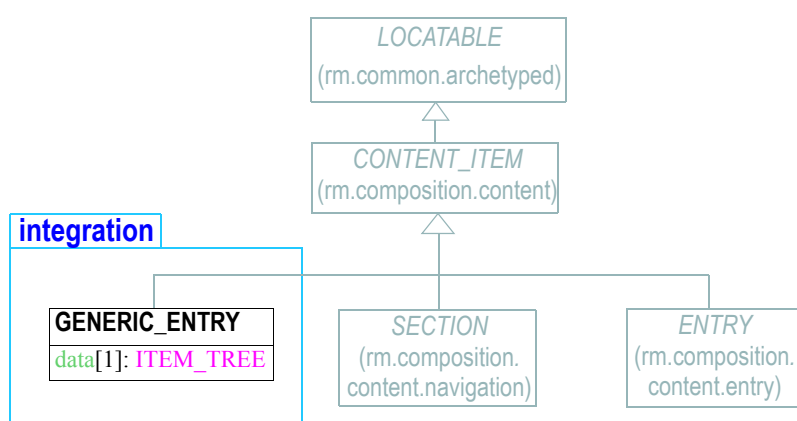


FIGURE 1 `rm.integration` Package

## 2.2.2 Semantics of `GENERIC_ENTRY`

A number of useful consequences follow from this modelling approach. Firstly, instances of `GENERIC_ENTRY` will contain attributes inherited from the `LOCATABLE` class, including *archetype\_node\_id*, and are thus archetypable in the same way as all other classes in the *openEHR* reference model. The `LOCATABLE` attribute *feeder\_audit* is also inherited, and may be used to mark every node of data with relevant meta-data from the source system record or message. Secondly, as a subtype of `CONTENT_ITEM`, `GENERIC_ENTRY` is a valid value for `COMPOSITION.content`. This is a completely desirable situation, since the same rules apply to `GENERIC_ENTRY` as to other content: instances can only be committed to the record as part of a `COMPOSITION` instance. `GENERIC_ENTRY` data are thus audit-trailed and versioned in the normal way. Thirdly, `GENERIC_ENTRY` instances can occur within a hierarchy of `SECTIONS`, which is useful for data sources which have headings or section equivalents (this is quite common in hospital information systems containing physician notes). Lastly, in common with all other *openEHR* data, design-time paths can be constructed for archetypes of `GENERIC_ENTRY`, while runtime paths can be extracted from data based on such archetypes. These path sets can be used for writing the data transformation rules.

It should be remembered that while `GENERIC_ENTRY` provides a standardised syntactic form for externally sourced data within *openEHR*, it provides no semantic coherence. This is particularly true for `GENERIC_ENTRY` instances sourced from numerous data sources: there is no guarantee that the `GENERIC_ENTRY` representations of “cholesterol result” from system A will be congruent with those sourced from system B. It is not even required that the data sources be vastly different for this prob-

lem to occur. Examples of messages can be found coming from different pathology laboratories, which obey the same minor version of HL7v2 (e.g. 2.3.1) and supposedly implement the same message type (e.g. “complete blood picture”) but which differ in actual structure and content.

The consequence of this situation is that `GENERIC_ENTRY` data cannot in general be safely used for clinical computation (e.g. decision support), and will not in general even support reliable clinical querying. In other words, a repository of `GENERIC_ENTRY`s (within appropriate `COMPOSITION` structures) does not constitute a reliable or interoperable health record - it can only be considered a standardised health information data store whose primary purpose is as the input to or output of semantic conversion processes, or for other auditing or non-clinical data management purposes.

### 2.2.3 Use with openEHR Extracts

The `GENERIC_ENTRY` class provides a way to represent data from non-*openEHR* systems that implement the *openEHR* Extract specification in order to either communicate with *openEHR* systems, or to communicate with other systems also implementing the *openEHR* Extract specification.

### 2.2.4 Integration with CEN EN13606

The `GENERIC_ENTRY` class provides a convenient basis for making *openEHR* systems EN13606-compliant, which in turn gives *openEHR* a gateway capability in heterogeneous environments where EN13606 is being used to communicate data. A CEN EN13606 EHR Extract can be converted to a series of `COMPOSITION`s containing `GENERIC_ENTRY` objects which obey appropriate integration archetypes; this data can then be semantically converted into orthodox *openEHR* objects for integration into a coherent EHR. Similarly, *openEHR* data can be converted into the `GENERIC_ENTRY`-based intermediate form for further conversion into EN13606 EHR Extracts.

## 2.3 Data Conversion Architecture

The integration archetype-based strategy for importing data into an *openEHR* system, shown in FIGURE 2, consists of two steps.

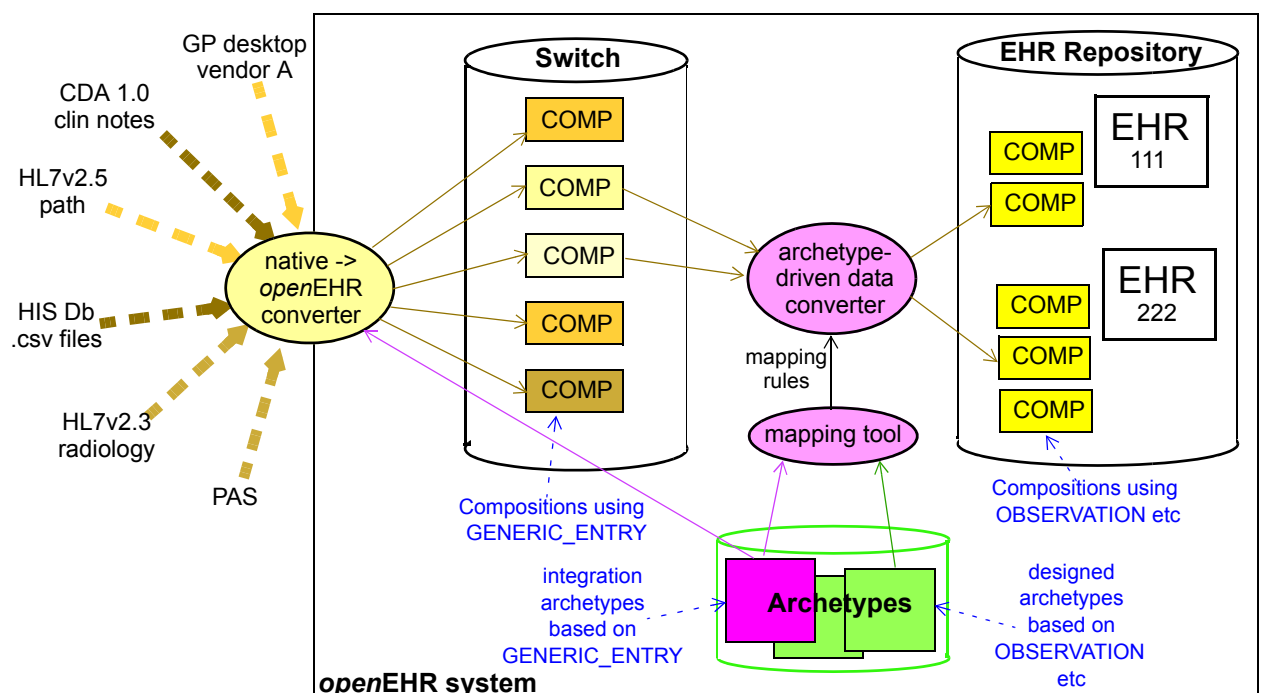


FIGURE 2 Data Integration using *openEHR*

Firstly, data are converted from their original syntactic format into *openEHR* COMPOSITION/SECTION/GENERIC\_ENTRY structures, shown in the *openEHR* integration switch. Most of the data will appear in the GENERIC\_ENTRY part, controlled by an integration archetype designed to mimic the incoming structure (such as an HL7v2 lab message) as closely as possible; FEEDER\_AUDIT structures are used to contain integration meta-data. The result of this step is data that are expressed in the *openEHR* type system (i.e. as instances of the *openEHR* reference model), and are immediately amenable to processing with normal *openEHR* software.

In the second step, semantic transformation is effected, by the use of mappings between integration and designed archetypes. Such mappings are created by archetype authors using tools. The mapping rules are the key to defining structural transformations, use of terminological codes, and other changes. Serious challenges of course remain in the business of integrating heterogeneous systems; some of these are dealt with in the Common IM document sections on Feeder systems.

## 2.4 Class Descriptions

### 2.4.1 GENERIC\_ENTRY Class

CLASS	GENERIC_ENTRY	
<b>Purpose</b>	This class is used to create intermediate representations of data from sources not otherwise conforming to <i>openEHR</i> classes, such as HL7 messages, relational databases and so on.	
<b>CEN</b>	Entry	
<b>Inherit</b>	CONTENT_ITEM	
Attributes	Signature	Meaning
	<b>data:</b> ITEM_TREE	The 'data' from the source message or record.
<b>Invariants</b>		

## A References

---

### A.1 Standards

- 1 ENV 13606-1 - *Electronic healthcare record communication - Part 1: Extended architecture*.  
CEN/ TC 251 Health Informatics Technical Committee.
- 2 HL7 version 2 ref....



**END OF DOCUMENT**