HL7 EHR TC

Service Functional Model Specification - Entity Identification Service (EIS)

Draft Standard for Trial Use, Release 1

December 2006

Publication of this draft standard for trial use and comment has been approved by Health Level Seven (HL7). Distribution of this draft standard for comment shall not continue beyond twenty-four (24) months from the date of publication. It is expected that following this twenty-four (24) month period, this draft standard, revised as necessary, will be submitted to the American National Standards Institute for approval as an American National Standard. A public review in accordance with established ANSI procedures is required at the end of the trial use period and before a draft standard for trial use may be submitted to ANSI for approval as an American National Standard. This draft standard is not an American National Standard. Suggestions for revision should be directed to Karen Van Hentenryck (KarenVan@hl7.org), HL7 HQ or via the HL7 DSTU Comments Web site at: http://www.hl7.org/dstucomments/index.cfm.

Service Functional Model Specification - Entity Identification Service (EIS)

Version 1.0

October 13, 2006

Project Lead

Barbara Eckman: baeckman@us.ibm.com
(IBM)

Editor

Alan Honey : alan.p.honey@kp.org
(Kaiser Permanente)

Authors

Virinder Batra: batra@us.ibm.com
(IBM)
Craig Bennett: cbennett@us.ibm.com
(IBM)
Ani Dutta: ani.dutta@va.gov
(Veteran’s Health Administration)
Dave Forslund: forslund@cognitiongroup.biz
(Cognition Group)
Alan Honey: alan.p.honey@kp.org
(Kaiser Permanente)

Preface

Notes to Readers

This document is the Service Functional Model for the Entity Identification Service, which is specified under the Service Development Framework process under the auspices of the Healthcare Services Specification Project (HSSP). Further context is given in the overview section below, but one key point to note is that the SFM provides a Service **Interface** specification, NOT the specification of a Service implementation. This is a critical distinction in terms of Service Oriented Architecture. There could be many different ways of implementing all or part of the functionality to support the behavior described in this specification.

Changes from Previous Release

This is the first public release of this document.

Acknowledgements

In addition to the listed authors, the following individuals are acknowledged for their contributions during the development of this specification.

Yongjian Bao: (GE Healthcare)

Grahame Grieve (Jiva Medical)

Don Jorgenson: (Inpriva)

Craig Parker: (Intermountain Healthcare)

Jari Porrasmaa, Juha Mykkanen and Marko Sormunen: (University of Kuopio / HL7 Finland)

Ken Rubin: (EDS)

Table of Contents

1 Overview

1.1 Introduction

1.1.1 HL7-OMG Healthcare Services Specification Project (HSSP)

The Healthcare Services Specification Project (HSSP) [http://hssp.wikispaces.com] is a joint endeavor between Health Level Seven (HL7) [http://www.hl7.org] and the Object Management Group (OMG) [http://www.omg.org]. The HSSP was chartered at the January 2005 HL7 meeting under the Electronic Health Records Technical Committee, and the project was subsequently validated by the Board of Directors of both organizations.

The HSSP has several objectives. These objectives include the following:

- To stimulate the adoption and use of standardized "plug-and-play" services by healthcare software product vendors
- To facilitate the development of a set of implementable interface standards supporting agreed-upon services specifications to form the basis for provider purchasing and procurement decisions.
- To complement and not conflict with existing HL7 work products and activities, leveraging content and lessons learned from elsewhere within the organization.

Within the process, HL7 has primary responsibility for (1) identifying and prioritizing services as candidates for standardization; (2) specifying the functional requirements and conformance criteria for these services in the form of Service Functional Model (SFM) specifications such as this document; and (3) adopting these SFMs as balloted HL7 standards. These activities are coordinated by the HL7 Services Oriented Architecture SIG in collaboration with other HL7 committees, which currently include the Vocabulary TC and the Clinical Decision Support TC.

Based on the HL7 SFMs, OMG will develop "Requests for Proposals" (RFPs) that are the basis of the OMG standardization process. This process allows vendors and other submitters (known as "RFP Submitters") to propose solutions that satisfy the mandatory and optional requirements expressed in the RFP while leaving design flexibility to the submitters and implementation flexibility to the users of the standard. HL7 members will be involved in the RFP creation and evaluation process.

It is important to note that the HL7 SFMs will focus on specifying the *functional* requirements of a service, while OMG specifications will focus on specifying the *technical* interface requirements of a service. In many cases, SFMs will also describe an overall coherent set of functional capabilities. These capabilities may be specialized or subdivided from both functional and informational (semantic) perspectives to provide specific "profiles" that may be used as the basis for the OMG RFPs and/or implemented.

1.1.2 Context of this SFM within HSSP Roadmap

As described above, the purpose of an HL7 SFM is to identify and document the functional requirements of services important to healthcare. Accordingly, this SFM seeks to define the functional requirements of an Entity Identification Service (EIS), which provides a set of capabilities to manage and retrieve identifying information for various kinds of entities (people, organizations, devices etc.). Once adopted as a HL7 standard, it is anticipated that this SFM will serve as the basis for one or more OMG technical specifications.

EIS will provide an important foundation component for many healthcare interoperability scenarios, both within and across organizations. Although in many business scenarios it may be used in conjunction with other services, it has been specified to provide stand alone capabilities.

1.1.3 Context and Relationship of this SFM within HL7

In terms of publishing, it is intended that this specification will appear in the "Services" section of the HL7 V3 publication, along with other services, such as RLUS and CDS. The relationship of service definitions to the message definition work traditionally carried out under the V3 banner will continue to be refined. This will involve joint sessions between the HL7 SOA SIG and MnM Committees and potential changes to the overall V3 guide. There are also organizational considerations for the new Transitional Technical Task Force (T3F).

In the mean time, this section identifies the relationship of this specific specification with HL7 V3 artifacts.

Services and Messaging offer an alternative paradigm for implementing similar functionality. It is beyond the scope of this document to consider the merits of each approach. However, both paradigms use information content structures known as "messages". Where both messaging and service based solutions are provided in a similar functional space, they must be based on the same conceptual information model. What may differ is the way that the information is "chunked" and the overall granularity of the message content. This alignment is achieved through the use of "semantic profiles" which are based on (i.e. use content extracted from) existing RIM based domain models. An example of this is given in Section 6 below.

This specification (along with other HSSP specifications) provides a separation of function from information content. This allows for different content models to be used within the same interface constructs. This has the advantage of enabling simpler version upgrades (e.g. when new RIM or domain model versions appear the functional specification does not have to change), but also allows organizations the flexibility of using semantic profiles based on other content models, e.g. HL7 V2 message constructs or non-HL7 content.

The term "Entity" and other related terms presented in the computational meta-model in Section 5 of this document are not RIM terms or as used in the RIM. The meta-model is not a RIM based model, since the information is for service configuration, not business data or to be used HL7 for messaging. In particular, the term "Entity" in this specification may refer to any "concept" or "thing" that may be identifiable and for which there is a requirement to resolve entities. This covers "things" such as People and Devices and concepts such as "roles" (Patient, Provider etc.). Since the purpose of the service is purely identification of the “Entity” any distinctions as to the nature of the entity are not important, other than obviously the actual data items used for that identification.

2 Service Overview and Business Case

2.1 Service Overview

2.1.1 Service Description and Purpose

The Entity Identification Service (EIS) Functional Specification is charged with defining the functional specifications of a set of service interfaces to uniquely identify various kinds of entities (e.g. people: patients, providers etc., devices) within disparate systems within a single enterprise and/or across a set of collaborating enterprises.

The following paragraphs and sections discuss usage of the service primarily with respect to patient identification, but similar functionality and scenarios are relevant to other entity types. The reason for concentrating on patients is both the familiarity and priority of the problem to a wide audience and also that initial profiles (defined in section 6) will be defined that support patient related information.

A typical person undergoes in his/her lifetime a vast array of healthcare encounters. Increasingly the nature of the encounter involves understanding the past experience and treatment of the patient, particularly in the case of chronically ill patients and in dealing with the large number of specialists that a patient may encounter. An accurate lifetime health record is becoming increasingly important in the overall management of the health of a patient. Also, throughout a person's lifetime he or she may have episodes of care provided by dozens or hundreds of healthcare providers, many of whom will assign and maintain patient IDs autonomously. In this arrangement, each organization or even department often assigns its own ID that uniquely identify the patient for its own purposes, with the result that these ID values are meaningless outside that system or organization. These autonomously managed IDs suit the purposes of recording and retrieval of service records for the single department or organization; however, there is no basis for efficient collection or correlation of health records among multiple venues or organizations.

The process of identification of a patient is well understood so that having a standard will improve the quality of care without compromising the role of innovation and discovery in healthcare. In addition to patients, providers and other entities or resources involved in patient care must also be uniquely and accurately identified.

This service is intended to allow for the resolution of demographics and other identifying characteristics (or traits) to a unique identifier. This allows any clinical system that uses the service to maintain a common description (set of traits) for each entity and to manage the entities. Having a standard interface for accessing and maintaining entity identification information allows systems and applications to have a consistent means of indexing data related to an entity.

By providing a standard mechanism for any clinical, financial, laboratory, or other application to uniquely identify and then retrieve an identifier associated with the entity, those applications may associate a variety of data with the entity identifier. By standardizing the interface to the service, clinical application vendors can leverage existing technology providers or open source implementations to more rapidly prototype and develop application and enterprise application integration infrastructures.

To put it simply, the Entity Identification Service provides the common thread by which entity data can be indexed. The unique identifier and standard way to search, retrieve and manage entity data will allow healthcare applications and healthcare enterprises to find, exchange and reference entity data while maintaining the data's context and associations.

2.1.2 Scope of the Service

The Entity Identification Service provides the functional definition of a service that can provide a robust and complete means for defining, updating and generally managing the entity identity, along with an associated set of identifying "traits".

The Entity Identification Service is intended to allow the lookup and management of a wide variety of entities including, but not limited to, patients, individual providers, institutional providers and medical devices. At the functional level, the service interface will explicitly allow identification of different "types" of entities that may be supported. Conformance profiles will be defined which may limit specific implementations to specific entity types and pre-define minimum trait sets for the individual entity types. This will also allow for performance optimizations to be defined for specific searches on specific entity types (which are implementation considerations which will be considered in the technical specification arising from the OMG RFP process).

The scope of this functional specification covers both support for multiple Entity Domains (see section 2.3.2) and multiple entity types. In both cases, it is an issue left to the issuers of the RFP and the RFP Submitters whether to define one or more technology specifications that provides an implementation specific to one type of entity and/or for a single Entity Domain. Both of these issues are discussed in more detail in Appendix IV.

2.1.2.1 Rationale

The main reason for defining this service at the more abstract "entity" level is that the interface functionality involved in identification of different kinds of entities is the same and that some EIS instances will be used to manage more than one type of Entity. The information model varies in that different characteristics or traits are used to search for the entities. The interfaces and operations defined in this specification are equally valid for patients, providers, devices and many other kinds of entities. From a technical or systems development perspective, this enables common frameworks and applications to be built and significant reuse to be achieved. From a business perspective, this provides greater flexibility and would allow re-classification of entities and roles without the need to change the service interfaces.

In many applications, the distinction between "entities" (real things, such as people) and "entity playing a role" (entities that are contextually defined by relationships, such as patient, provider, member, customer etc.) is significant. However, with respect to identification in this service the distinction is not significant. The remainder of this specification will use the term entity to imply either an entity or an "entity in a role". A side effect of the abstract approach is that it would be feasible to use the "linking" functionality defined in this service as a means to link roles to the entity playing them. This, however, is not a primary use case or intention of the service.

The determination of the exact usage is dependent on the Semantic Profile (see Section 6) being used. From the point of view of the Service, there is no difference, i.e. the Service can be used to identify "people" or "patients" equally, depending upon which Semantic Profile is being used.

The scope of the service may also extend across multiple Entity Domains (see Sections 2.3.2, 2.4, 3.2.2 and Appendix IV for further discussion).

2.2 The reason why the service is necessary.

A typical healthcare information system should permit the user to submit a search for a person’s medical record using some combination of identifying parameters for the person. When the user must collect a patient's healthcare information from a different organization or from a disparately-keyed system in the same organization, s/he typically must perform a new search in that other system - or ask a medical records person in the other organization to perform the search - in order to identify the person and retrieve the needed information. This is the basic need for this service.

The following business and technical considerations provide further rationale and justification for this service:

1. In order to be able to provide a lifetime health record and continuum of care across multiple evolving organizations and venues, a robust standard mechanism to identify an individual as being the same person with differing identifiers at multiple organizations is needed. This is required even if a universal person identifier were used, since one must have a process to link the identifier with an individual.
2. The need for a standardized set of person identifiers.
3. A common mechanism is needed to be able to manage the identifiers of a patient and determine with high confidence that a person is who they are reported to be across multiple enterprises simultaneously. With the development of Regional Health Information Organizations, the need to provide a distributed Electronic Health Record (EHR) has been identified and an accurate identification mechanism of patients is a critical component to delivering such an EHR. At the heart of an Electronic Health Record (EHR) system is some means to uniquely identify patients and other entities that interact with the healthcare system.

In addition to the business need for this specification to support business processes, the specific scope and functional definition of this specification will enable technical specification and implementation as a web service (and also other potential technical implementations

Working with the EIS specification will reap the following additional benefits for its stakeholders:

1. The ability to provide and utilize a common infrastructure for identifying persons can make it easier for a vendor to bring a product to market and provide functionality that might be greater than having to deliver a standalone system which would include this capability. Vendor competition is an opportunity to provide more efficient and useful implementations of a standard specification. The provider of this specification might have a larger market than they would otherwise as the use of the system may become much broader than in a standalone EHR system.

2. While many healthcare services and application vendors provide some form of Master Patient Index (MPI), the level of functionality and the implementation details vary greatly. Vendors currently provide MPI functionality because the ability to uniquely and consistently identify patient is necessary to integrating any systems where patient data is stored. The functionality of an MPI is supportive and not a core function of the clinical, financial, laboratory, or other applications in the healthcare setting. An MPI implementation is generally not considered a product differentiator. A consistent interface to an Entity Identification Service could allow vendors to concentrate on the business issues for the applications and take advantage of best-of-breed technology for entity identification. For those vendors that supply best-of-breed MPI functionality, providing standard interfaces to their MPI by way of the Entity Identification Service specification would allow those vendors to provide their MPI as key piece of middleware for clinical information system vendors.

3. Consumers of healthcare applications would benefit from the ability to select an Entity Identification Service that meets their needs. In addition, as business environment changes dictate, a standard set of interfaces would allow application consumers to change the Entity Identification Service provider without impact to existing applications

4. From the point of view of a hospital system or a department of health there are potential large cost savings that could be achieved by being able to utilize a common framework for person identification that would work across different commercial products, each of which might be specialized to provide health information specific to its scope. The ability to competitively procure such an identification component could potentially reduce the cost of such a system. For the patient, the benefits would be the ability to receive a higher quality continuum of care across multiple provider and care organizations. This might also be able to impact the medical insurance industry by reducing costs in the process of ensuring accurate care of the correct patient.

2.3 Structure of the Service

2.3.1 Interfaces

This specification supports four identified interfaces: Administration, Service Metadata Management, Entity Identification Management and Query.

- Administration supports operations for technical support (start, stop etc. also includes ability to permanently remove entity records). Only the “Remove Entity” capability has been included in this SFM. All other administrative functions would be defined in the Technical specification (OMG RFP Submissions)
- Service Metadata Management operations focus mainly on managing the metadata that an individual instance of EIS supports (see section 5 for further discussion).
- Entity Management provides operations for manipulation of Entity Identifiers and traits.
- Query provides query operations for discovering entity identifiers and traits.

Each of these interfaces and their operations are described in Section 5 below.

Since this specification provides capabilities for managing identifiers for generic “Entities”, there will be a need to specialize the information model for specific types of Entity, such as “Person”, “Patient” or “Provider”. The HSSP method for achieving this specialization is to define “Semantic Profiles”, which are identified in section 6. This specialization will consist of defining a pre-determined set of traits that an instance of EIS supporting the profile must support. This version of the specification only includes one such profile, i.e. “HL7 V3 Patient V1.0” (See section 6). Note that as well as the identification of the traits, this necessitates that certain parts of the service metadata must also be included in the profile.

It is envisioned that many other profiles will be defined at later stages. Additional profiles may also be defined by Submitters during the RFP process.

HSSP services also use the concept of “Functional Profiles” which may allow for different subsets of the interfaces to be provided as coherent named sets. This specification identifies a base set of functional profiles, and also identifies certain operations as optional with respect to the RFP submissions (since the same functionality can be provided by other operations, e.g. individual retrieves as well as “list” style query functions).

2.3.2 Use of Entity Domains and EIS vs. XEIS

A key structural concept defined within this specification is the “Entity Domain”. Within this context, this identifies a sphere of use of entity identifiers.

Note that this is not necessarily the same as the actual sphere of “control” of the identifiers in terms of their issue. The sphere of issue control could be legal (e.g. government issued identifiers), organizational (e.g. department, enterprise, cross-enterprise), geographical (e.g. regional, national, state) or even specific to one computer system. The “sphere of use” is more the set of organizations that wish to interoperate for a specific set of information. For example, a Social Security Number may be issued (and controlled) by a central federal authority. It can then be used as an identifier for “patient” in one Entity Domain, and used as an identifier of “employee” in another Entity Domain. In this case, the organization is “controlling the usage” of the identifier. The discussion in the remainder of this document focuses on the “usage” aspects only. The combination of “usage” Entity Domain and Entity Type should be sufficient to resolve any ambiguity, given that Entity Type in this context could identify either an Entity or an Entity in a Role (i.e. will identify a "Company X Employee" or a "Company Y patient" or a "Company Z NorthWest Region patient").

The Entity Domain is the means by which the usage of identities is managed across different departments or organizations. An individual EIS instance could support multiple Entity Domains, but any individual interaction with an instance of EIS always occurs with respect to one specific Entity Domain (other than explicit linking of entities across Entity Domains). Allowing a service instance to support multiple Entity Domains enables the right level of encapsulation or abstraction from organizational structure, software topology and distribution. This seamless design characteristic is one of the primary benefits of a Service-Oriented Architecture.

However, it is quite feasible and reasonable to expect that some EIS instances will only operate across one Entity Domain. This is depicted in the diagram below:

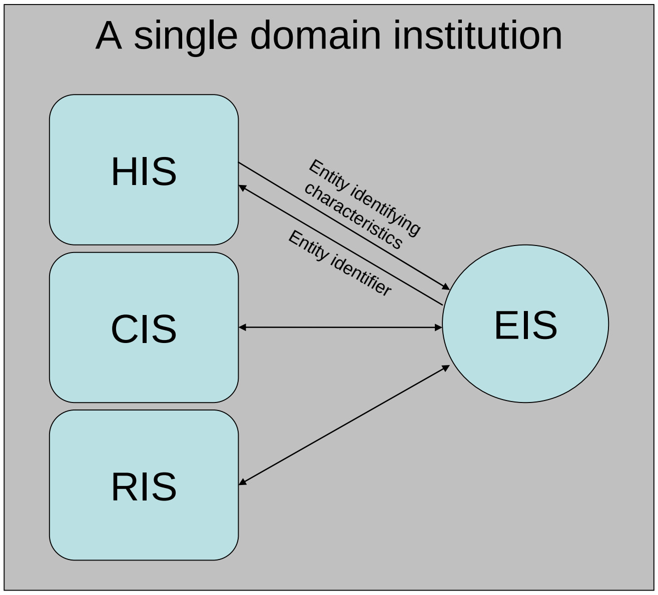

A single domain institution
HIS
CIS
RIS
EIS
Entity identifying characteristics
Entity identifier

Figure 1 - Single Entity Domain EIS

It is also expected that many EIS instances will cooperate in forming a hierarchic or peer-to-peer network pattern in order to satisfy requests. EIS instances that operate across more than one Entity Domain can be considered to be "Cross Domain Entity Identification Services" (XEIS). The XEIS can provide the ability to manage identifiers within multiple Entity Domains and relate multiple local Domain Entity identifiers to a single cross domain entity. For patient identities, the XEIS can be compared more to an EMPI (Enterprise Master Patient Index) in terms of scope and functionality. A sample set of relationships between EIS and XEIS instances is depicted below:

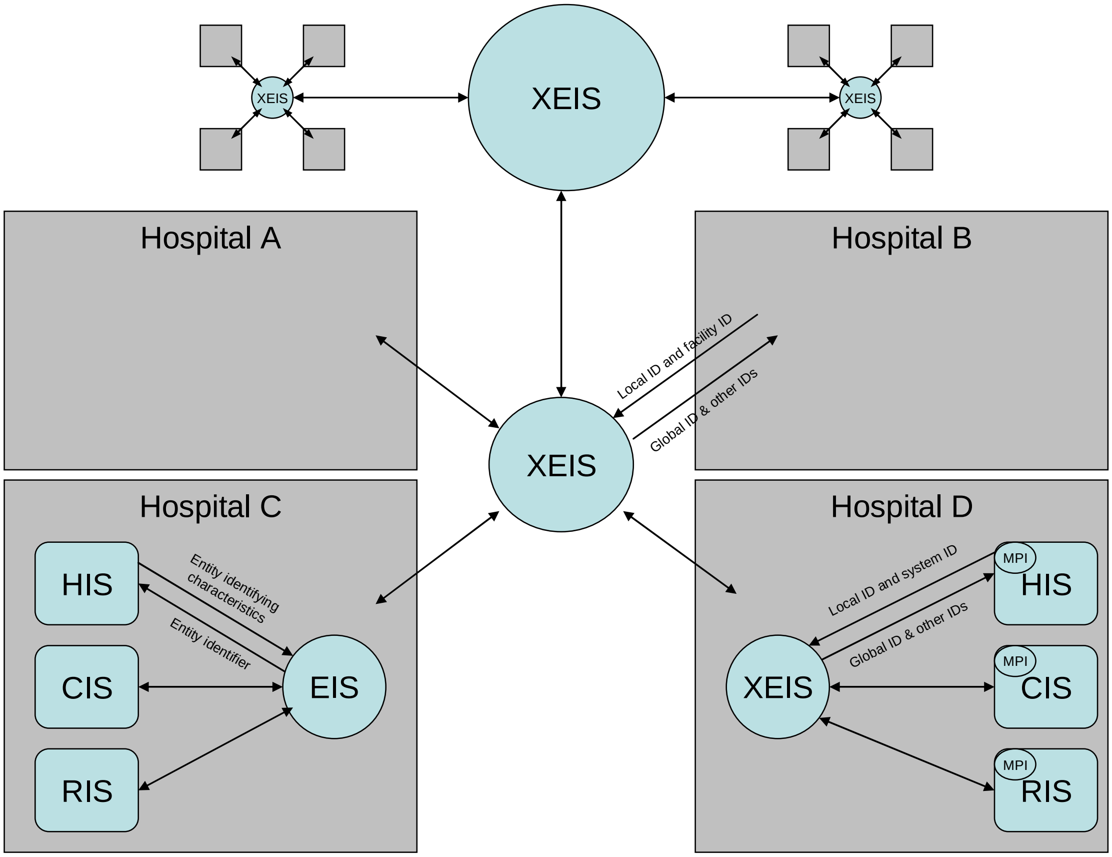


Figure 2 - Multiple Entity Domain EIS (XEIS)

In addition to the EIS capabilities of finding an entity based on identifying traits (a.k.a. attributes, a.k.a. characteristics) and the management of the entities and the traits, the XEIS must also allow one or more Entity Domain specific identifiers to be associated with an entity and provide the access and management capabilities for those local Entity Domain identifiers.

In this context, a local Domain Entity identifier is any identifier that has been allocated by an individual system or facility. Note that the facility or system itself would have been issued its own unique identifier within the context of the XEIS.

Where the distinction is relevant, the remainder of the document will use the terms as follows:

EIS - Entity Identification Service that operates in a single domain only

XEIS – EIS where multiple domains are recognized and managed.

In other sections, which are referring to general capabilities of the specification or overall description, EIS will be used.

2.4 Implementation Considerations

Functionally, the interfaces for EIS and XEIS are similar, the only differences being that some metadata operations are specific to XEIS instances and also that some XEIS operations can deal with different Entity Domain values. It is a consideration for the technical specification and implementation how to implement the Entity Domain concept for both XEIS and single domain EIS instances. Since it defines the “super-set” of functionality, the inputs for all interfaces in this document are defined as if the implementation was a fully functioning XEIS. Using profiles or otherwise, the RFP submissions may apply constraints to provide the subset of the functionality provided by more specific EIS interfaces. This is discussed in detail in Appendix IV.

As well as considerations of Entity Domains, architects will need to consider the many approaches to providing the ability to resolve identifiers for different entity types such as individual people, devices and facilities. For example, a priority requirement may be the need to provide a high throughput interface that is specialized toward resolving entities that have a name and address. There may also be a need to make optimizations associated with population demographics (see also later discussion in section 4.2). This could suggest special circumstances regarding nicknames, common roadway names, and geospatial issues that could benefit from special handling of the class of entities that contain people and facilities that may relocate of their own volition or change or abbreviate names.

The technical specification (RFP submission) will need to consider whether there may be a need for additional entity-type specific operations over and above the generic ones. Where an EIS or XEIS instance supports multiple types, one option may be to implement the special processing as private operations within the service implementation, with a simple switch based on entity and request type from the generic interface. Another option could be to define specific profiles that provide matching operations dedicated to a specific entity type.

The use of profiles is key in providing real implementations of this service. These are discussed in various sections within this document. It is important to note that for those wishing to implement a “simple” EIS, e.g. basic identity resolution for Patients only, then this will be easy to do by applying constraints such as profiles. This is discussed further in Appendix V.

As an interface specification, EIS may be implemented and used in various different topologies. It may be access directly via web or other User Interface (UI) mechanisms by users, or system-to-system via other applications such as Hospital Information Systems. This does not affect the interface or behavior specification.

Also, whether the implementation behind the EIS interface is actually the source of record or just a separate indexing mechanism is again an implementation choice. However, it should be able to be trusted as an authoritative source, but again this does not affect the interface or behavior.

3 Business Scenarios

Note that the term Hospital Information System (HIS) is used as a generic term for various different types of clinical systems, including Patient Administration Systems. The nature of the system that acts as the client of the EIS does not affect the service definition.

3.1 Primary Actors

- Robert (Bob) Smith – a patient
- Mary Jones - Patient
- Evelyn Smith – a patient
- Greg – a hospital registration clerk
- Fred – a hospital pre-op nurse
- Martha – a discharge nurse
- Carol – an EIS administrator
- Peter - CIU technologist
- Sarah - HIS record manager
- William – a laboratory clerk

3.2 Primary Scenarios

3.2.1 Single Entity Domain Scenarios

The following scenarios describe a number of real-world activities around patient identity management in hospital Moon. The hospital uses one single Entity Domain to manage the patient identities. Medical records are administered by its Hospital Information System (HIS). The HIS actually manages the medical records and encounter information. An Entity Identification Service (EIS) is used to manage and access patient identities across all of the hospitals systems. (Note that the actual system providing the implementation of the service interface could be part of one of the clinical systems or a separate EMPI system). In the scenario descriptions, the term EIS is used to represent both the service itself and the system that provides the implementation of the service. All other care information systems use the same Entity Domain to identify patients. Although the scenarios depict “patients” as the entity type, similar scenarios can be implied for other entity types, such as Providers or other resources. Although these have not necessarily been strung together to provide a single coherent story line, it is fairly easy to see how some of them can be composed.

3.2.1.1 Create a new patient

Robert (Bob) Smith enters the hospital for outpatient surgery. Upon arrival in admitting, Greg, a registration clerk, attempts to find Bob in the HIS using Bob’s name and address. Bob is not found locally in the HIS, so the HIS uses the EIS to try to locate Bob, but he is not found. Greg asks if Bob had ever been to the facility. Bob thinks he has and Greg asks if he had moved. Bob gives Greg his old address, but Greg is still unable to locate Bob's information in the HIS or EIS.

Greg treats Bob as a new patient in the facility and gathers all of Bob's demographic information. Bob's patient record is now created in the HIS and appropriate identifying information is passed to the EIS for creation of an new entity. Bob is through with the admission process in the HIS, and an outpatient encounter is created in the HIS.

3.2.1.2 Link (or Merge) entities

As Bob is checking out of the hospital, he once again provides his identify information to Martha, a discharge nurse. When she enters the information and searches for Bob in the HIS, she sees two entries that look very similar. Martha asks Bob if he had ever lived at the address of the former record; when Bob responds positively, Martha more carefully inspects the information and finds that it is actually another entry for Bob. Martha selects both records and requests the entries to be linked. The HIS requests the EIS to link the records and the newer entity record becomes an alias for the older record.

Note: Depending on the hospital's patient information administration policy, Martha may instead request for EIS to merge the two records, and update some traits (e.g., the address) of the survived record (which represents the resultant entity record of the merge operation). If the hospital chooses merge operations to resolve duplicated entity records, the merged record becomes deprecated after the merge, and only the survived record is used to identify the entity in the future.

3.2.1.3 Update demographics

The next time Bob enters the hospital, Bob gives Greg his name and address, but Greg cannot find an exact match in EIS with the information provided. After asking for Bob's date of birth, the set of near match records that were returned from EIS are examined and the entry with the top matching confidence score in the list appears to be a very close match for Bob, but the address does not match.

After retrieving the full record for this entry from the HIS and asking Bob some further qualifying questions, Greg realizes that Bob has moved since his last admission and discharge. Greg updates Bob's address and phone number in the demographics and submits the change to the EIS.

3.2.1.4 Inactivate entity from general searches

According to company policies, Carol discovers that Bobs record should be inactivated. (The circumstances for inactivation would be defined by organizational policies, but could include death, relocation, change of health plan, removal of test data from a production system etc). While Bob's records still must be accessible, his demographic information should not be considered when looking for active patients. Carol, an administrator, uses Bob's identifier in the hospital's local Entity Domain to locate his entity record in the EIS, and changes its state to Inactive. After this change, Bob should not be considered in any searches for clinically active patients. This does not preclude

Carol or any other administrator from finding Bob’s entry using administration tools of EIS.

3.2.1.5 Activate (inactive) entity to general searches

Carol realized she had made a mistake in removing Bob’s records from active searches. Whatever the circumstances that led Carol to inactivate the record were discovered to be incorrect. Carol, using her EIS search capabilities, found Bob’s entry, selected the entry and changed its state to Active in the EIS, to allow Bob’s records to be found by general clinical searches.

3.2.1.6 Unlink (unmerge) entity

Miss Evelyn Smith enters the hospital and gives Greg her information. Greg is able to find an Evelyn Smith in the HIS system, but not the correct Evelyn Smith. Miss Smith is adamant that she has been in this particular hospital several times in the past. As Greg investigates, he finds that there are admission records corresponding to this Miss Evelyn Smith’s hospital visits currently associated with Mrs. Evelyn Price. Digging deeper, Greg finds that Mrs. Evelyn Price was recently married and her maiden name was Smith.

Realizing that Mrs. Evelyn Price’s entry had become linked to Miss Evelyn Smith, Greg removes the alias from Miss Evelyn Smith’s entry in EIS so that there are now three distinct entries: Miss Evelyn Smith, Miss Evelyn Smith, and Mrs. Evelyn Price. The clerk is now able to find and select the appropriate Evelyn Smith from the list and proceed with the admissions process.

Note that in the case of “merges”, there will be two options for unmerging, one being an automated “undo” type capability, which is only applicable if no further changes had been applied to the records, the second being an “unmerge” which requires specific manual action.

3.2.1.7 Look up a patient

3.2.1.7.1 Single Entry found

After Bob has been admitted, he is taken to a pre-op waiting room. In the pre-op waiting room, Fred, a nurse, asks Bob for some identifying information and looks up Bob’s records in the Hospital Information System (HIS). Since the EIS is shared, Fred finds Bob’s information to locate his encounter. Bob’s vital signs and other clinical data are recorded in the HIS, under his encounter.

3.2.1.7.2 Multiple Entries found

Robert (Bob) Smith enters the hospital for an outpatient appointment. Greg, the registration clerk, asks Bob for his name, and performs a search on his name in the HIS. The HIS asks the EIS to perform the search, and Greg finds several hundred matches for some variation of Robert Smith. Greg asks Bob for his age and some address details, and then repeats the search. Of the few records returned, Greg decides that one of the records is a match for Bob, and the HIS presents the records for that identity. Greg creates a new outpatient encounter record in HIS.

3.2.1.7.3 Merged Entries found for an Identifier

William is adding a lab test to the lab system and used an identifier that has been previously merged into another which has been designated as the “main” identifier. The lab system does not recognize the identifier so checks with the EIS which returns an indication that the record has been merged, together with the main active identifier.

3.2.1.8 Unattended encounter

The referral system receives an HL7 message from another institution describing a referral for a patient that is about to admitted. The referral system does not recognize the patient, so it passes all the information it can find and asks the EIS to identify the patient.

3.2.1.8.1 EIS response #1

The EIS finds a record matching the patient details and returns the identifier. The referral system can complete processing the referral.

3.2.1.8.2 EIS response #2

The EIS finds more than one record meeting it's criteria for unattended patient encounter matching, and no record meeting it's criteria for a complete match. It returns a list of possible patient identifiers. The referral system cannot complete processing the referral and a user must look at the referral to determine the correct identifier.

3.2.1.8.3 EIS response #3

The EIS finds no potential matches. The rules within the referral system are configured to request a new identifier to be created and then to send a request to the EIS to do so. The EIS creates the identifier and returns it to the referral system so that it can complete processing the referral.

3.2.1.9 Remove entity from the system

While testing a new function in a system that uses the MPI in production, Carol created a pseudo-record for a fictitious patient. After Carol had completed all testing, Carol found the record, selected the record and removed the record from EIS. Once Carol confirmed the entity removal request, she permanently purges the record.

Note: Though this record may be kept in some repository for audit trail purpose, nobody, neither clinical user nor administrator, can find it from EIS.

3.2.2 Multiple Entity Domain Scenarios

The following scenarios introduce cross Entity Domain processing. In addition to hospital Moon introduced above, these scenarios consider a separate multi-layer organization, Solar System Healthcare, with 4 Regions, East, West, North and South, each of which manage their own identifiers in their own EIS instances. Hospitals Jupiter and Saturn are assumed to be in different regions (North and East respectively) within the organization.

Solar System operates a “master” national XEIS with its own national identifiers, against which the regional identifiers are cross-referenced.

Another single hospital healthcare organization, Orion is also referenced for cross organizational scenarios. In addition to the actors listed above, the following actors are referenced:

- John – a hospital registration clerk in hospital Orion
- Frank – a Solar System patient in their East region
- Celia – an Orion patient.
- Pablo – a hospital registration clerk in hospital Jupiter

3.2.2.1 Look up a patient across a regional network

Bob, who normally receives his healthcare from hospital Moon, is admitted to hospital Orion while on vacation suffering from chest pains. John does not find record of Bob in their EIS system. He asks Bob for some identifying information and requests their EIS system to pass the search to their RHIO network to see if Bob can be found anywhere in the network. Hospital Moon is also connected to the RHIO network and its EIS finds a match with the details supplied and returns their identifier for Bob. This is then used to retrieve Bob’s medical record from Hospital Moon (either by use of an RLUS or by Fax or other mechanism).

3.2.2.2 Look up a patient specifying a specific external organization

Celia is admitted to hospital Moon suffering from an ankle injury. She informs Greg that she has never had treatment from hospital Moon but that she is a patient at Orion. Greg enters a request to EIS to locate Celia entering her name and address and indicates that she is an Orion hospital patient. The EIS contacts the RHIO network specifically requesting a search in Orion’s EIS. Orion’s system locates her record and returns her identity. This is then used to retrieve her medical record from Hospital Orion (either by use of an RLUS or by Fax or other mechanism).

3.2.2.3 Link entities across regions within an organization

Frank is admitted to Jupiter hospital (in the North region of Solar System Healthcare). Pablo checks for Frank on their HIS system but he is not found. Frank then informs him that he is a Solar System patient in another area of the country, so Pablo requests the XEIS to search the whole of Solar System Healthcare. Frank’s details from the East region are found by the National XEIS and his details are returned. A new identifier is created for Frank in the North region so that the HIS can process his record and a link is created in the Solar System’s master XEIS to tie Frank’s new regional identifier to the master identifier for Solar System. (Note – the Master XEIS could hold a duplicate of all regions’ entity traits or pass the request to retrieve them to the East region EIS, this is an implementation choice)

3.3 Supplemental Scenarios

None.

4 Service definition and dependencies

4.1 Service Definition Principles

The high level principles regarding service definition that have been adopted by the Services Specification Project are as follows:

- Service Specifications shall be well defined and clearly scoped and with well understood requirements and responsibilities.
- Services should have a unity of purpose (e.g., fulfilling one domain or area) but services themselves may be composable.
- Services will be specified sufficiently to address functional, semantic, and structural interoperability.
- It must be possible to replace one conformant service implementation with another meeting the same service specification while maintaining functionality of the system.

[use SFM]A Service at the Functional Model level is regarded as a system component; the meaning of the term “(system) component” in this context is consistent with UML usage[1].A component is a modular unit with well-defined interfaces that is replaceable within its environment. A component can always be considered an autonomous unit within a system or subsystem. It has one or more provided and/or required interfaces, and its internals are hidden and inaccessible other than as provided by its interfaces.

Each Service’s Functional Model defines the interfaces that the service exposes to its environment, and the service’s dependencies on services provided by other components in its environment. Dependencies in the Functional Model relate to services that have or may in future have a Functional Model at a similar level; detail dependencies on low-level utility services should not be included, as that level of design is not in scope for the Functional Model.

The manner in which services and interfaces are deployed, discovered etc is outside the scope of the Functional Model. All other interactions within the scope of the scenarios identified above are in the scope of the Functional Model.

[1] It is expected that services will be defined, in response to the OMG RFP process, as UML components, however that level of design is outside the scope of the Functional Model.

Reference may be made to other specifications for interface descriptions, for example where an interface is governed by an existing standard.

4.2 Overall Pre-Conditions, Dependencies, and/or “Out of Scope Statements”

While a generalized interface for entity resolution has been defined in this specification, the differences among types of entities require specialization at the implementation level, and possibly additional specialized operations. While a device tends to be immutable, a person has a constantly changing set of attributes.

Because of these differences, techniques and algorithms for determining the uniqueness of an entity may differ depending on the type of entity being described.

For example, a person generally has a first name and last name. These attributes are important in resolving the identity of a person. However, a person may have a first name of Richard, but in varying stages of the person’s life, they may use a moniker of Rick, Ricky, Dick, Rich, or Ritchie. A person may also marry or legally change their name for some other reason. Knowing that an entity is a person and knowing common variations on these attributes allows for more intelligent entity identity resolution than simple string comparisons.

On the other hand, a device entity may always have a ‘manufacturerModelName’ that would be inappropriate for a person, but may be considered a significant attribute when attempting to resolve the identity of a device.

In resolving entities based on attributes (a.k.a. traits), in addition to required, easily identifiable attributes and the potential special handling of a specific subset of those attributes, the significance of certain attributes in identity resolution help with the ‘weighting’ of the possible match candidates. For example, if a person enters a clinic and provides the attributes: “name=’Bob Walsh’” and “addr=’123 Main Street’”, the EIS implementation may weight a match with an existing entity with attributes: “name=’R Walsh’” and “addr=’123 Main Street’” higher than an existing entity with attributes: “name=’Bob Walsh’” and “addr=’123 Elm Street’”.

A service provider that specializes in resolving devices, on the other hand, may have a deep knowledge of specific device manufacturer conventions on serial number generation, model names and numbers and/or technical specifications.

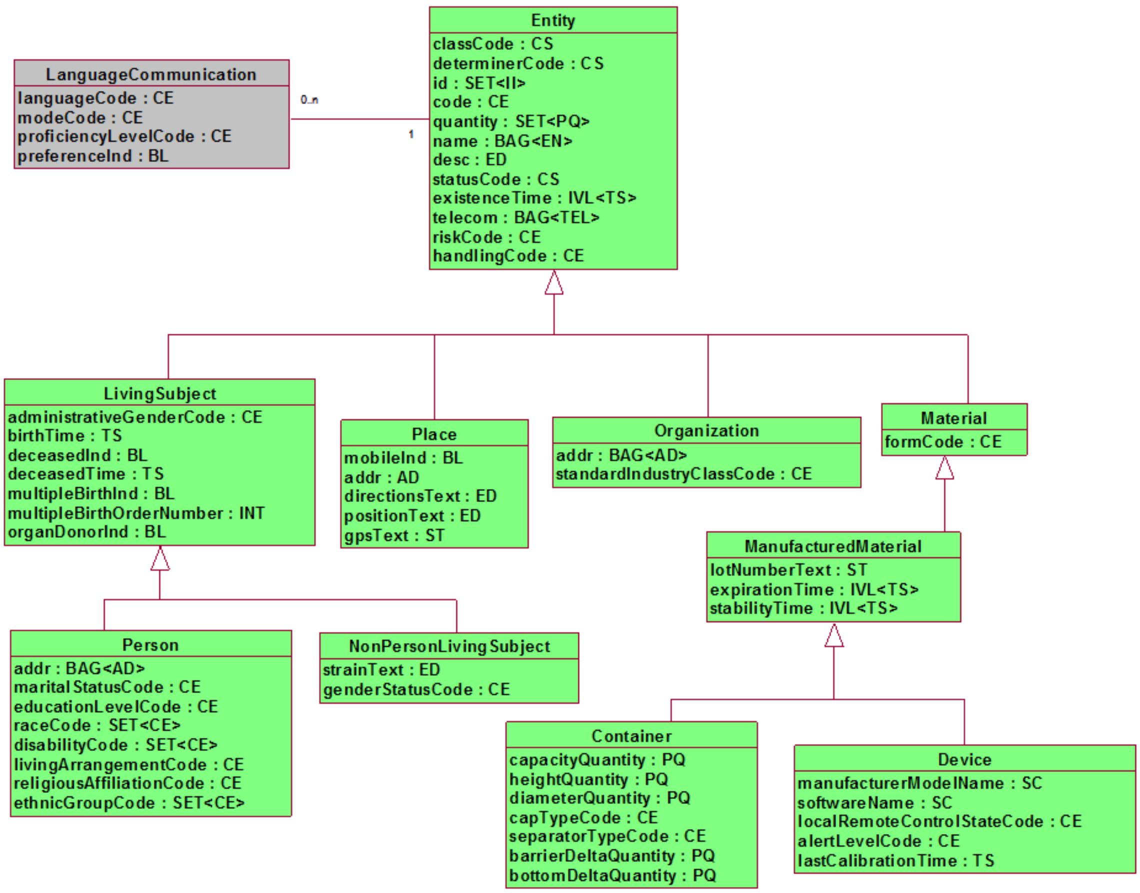


Figure 3 - HL7 RIM

As can be seen from the entity specification from the HL7 RIM, entities are uniquely defined with distinct attributes based on subclass. This entity uniqueness must be supported by the EIS service interface. This will be achieved through the definition of Entity Types, as described above and then by defining standard traits that are mandatory or optional for each type.

This functional model provides generic interface capabilities to add and maintain entities and retrieve identifiers etc. This allows for definition of different Entity Types and associating Traits with those Entity Types. The definition of minimum/mandatory trait sets and their relationship to specific Entity Types will be handled using “Semantic Profiles”. One initial profile has been defined, i.e. “HL7 V3 Patient”. It is expected that HL7 domain committees or other organizations may define additional specific profiles. This issue may also be addressed during the RFP issuance and submittal process.

Also, this specification does not constrain whether the EIS interface is provided to an "authoritative" source of information or not. EIS interfaces may hide such a master system, or they may be used just for registry purposes, while authoritative information is kept in other systems (e.g. HIS).

5 Detailed Functional Model for each Interface

This section gives the functional description of the interfaces for this service. Four separate interfaces have been defined:

- Administrative Functions: These functions are expected to be used only by systems administrators who maintain the Information Technology Infrastructure.
- Service Metadata Functions: These allow for maintenance of the metadata used by the EIS, including traits and entity types.
- Entity Identification Management Functions: Provides operations for manipulation of Entity Identifiers and traits.
- Query Functions: Provides query operations for discovering entity identifiers and traits and metadata discovery. Also includes the capability to query which conformance profiles (see section 6) an EIS instance supports.

Note – the term "Entity" is used throughout the interface and operation descriptions. The EIS only accesses and references certain identifying information (traits) about the Entity, acknowledging that the main information about the Entity will be stored elsewhere (potentially accessible through other services such as the HSSP Resource Location and Updating Service - RLUS), but for simplicity the term "Entity" is used within EIS, e.g. as in "Create an Entity".

The functions specified in this document will use only demographic data as these functions are meant to find the identity of the entity we are looking for. Data about the entity is entered in the individual systems which collaborate with an Entity Information system supported by the regional Enterprise.

It is also expected that there will be a 1-to-N mapping of scenarios to these functions where each scenario may use multiple of these functions to satisfy the scenario.

This specification uses a number of concepts to achieve the balance between flexibility and usability. A computational meta-model is given below. Note that data types on attributes are only indicative and are not a constraint on the technical specification.

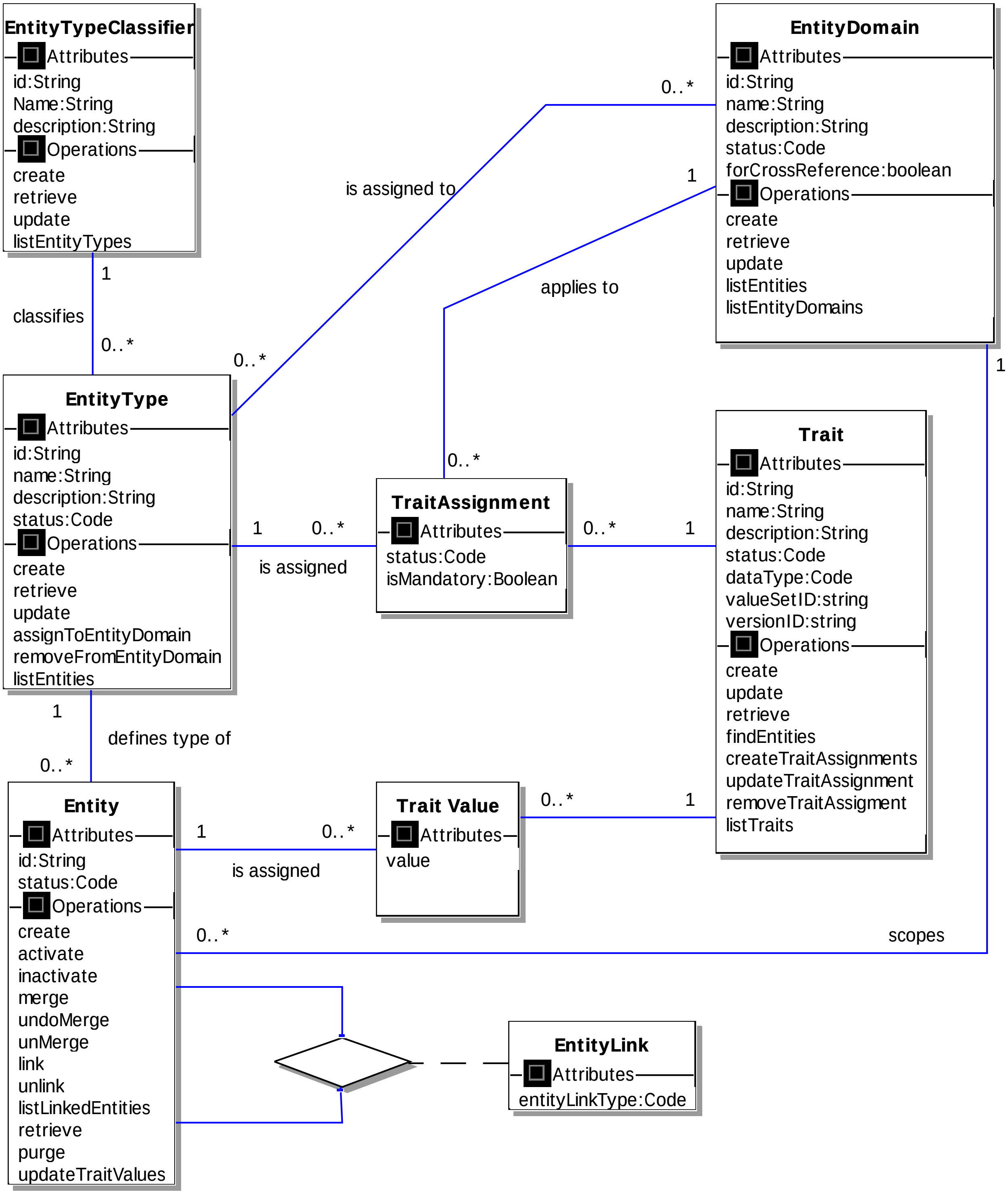


Figure 4 - EIS Computational Model

- Entity Domain – this is similar to the concept of security domains, and identifies the scope of usage of an Entity identifier. Entity Domains may be organizational or geographical. Each Entity Domain must have an owner that can define policy and which entity types are supported within the Entity Domain. Attribute descriptions:
    - id – unique identifier of the Entity Domain
    - name – the name of the Entity Domain
    - description – a description of the Entity Domain
    - status – the status of the Entity Domain. Indicates whether the Entity Domain can be used.
    - forCrossReference – indicates whether this Entity Domain can be used in cross references of Entities
- Entity Type Classifier – this identifies a "category" of information that may be supported in an EIS. This can be defined at various levels and even customized for a specific organization (subject to interoperability concerns). Example values could be "RIM V02-14 Entity" or "RIM V02-14 Role". Attribute descriptions:
    - Id – unique identifier of an Entity Type Classsifier
    - Name – the name of the Entity Type Classifier
    - Description – a description of the Entity Type Classifier

  (Future versions of this specification may allow for further layers, thus providing a hierarchic classification scheme. From a meta-model perspective, this would simply mean adding a recursive relationship on the Entity Type Classifier, but due to the potential implementation complexities, this is not included in this version.)
- Entity Type – this identifies a specific "type" of entity that an EIS supports within specific Entity Classifiers, so for "RIM V02-14 Entity" Type Classifier, the types could be: Person, Living Subject, Organization, Place, Material etc. Attribute descriptions:
    - Id – unique identifier of the Entity Type
    - Name – the name of the Entity Type
    - Description – a description of the Entity Type
    - Status – the status of the Entity Type. Indicates whether the Entity Type can be used, e.g. whether new entities of the type can be created.
- Trait – this is an individual characteristic that is applied to a specific Entity Type for use within a specific Entity Domain. Examples for Person would be: name, address, SSN etc. Note that the valueSetID and versionId would only be used where a trait is a coded value, and would identify the code or value set to be used. Attribute descriptions:
    - Id – unique identifier of the Trait
    - Name – the name of the Trait

  - Description – a description of the trait.
  - Status – status of the Trait. Indicates whether the Trait can be used for Trait Assignments.
  - dataType – the data type of the Trait. (For HL7 profiles, this will be a RIM Data Type)
  - valueSetID – identifier of the value set (if applicable) which defines the allowable values for this trait.
  - versionID – identifier of the version of the Trait.
- TraitAssigment – this allows traits to be assigned to Entity Types, but also to vary this assignment for different Entity Domains. The Boolean isMandatory attribute determines whether the Trait has to be present or not for the given entity type within the given Entity Domain. Attribute descriptions:
  - status – the status of the Trait Assignment. Indicates whether it is currently in use for the Entity Type.
  - isMandatory – indicates whether the trait is mandatory for the Entity Type and Entity Domain.
- Trait Value – this is the actual value of a particular trait on a particular entity instance, e.g. “Smith” for the Trait “last name”. Attribute descriptions:
  - value – the actual value of a specific trait for an Entity instance
- Entity - represents the actual instance entries in EIS. These are always of a specific Entity Type, and are scoped within a specific Entity Domain. Attribute descriptions:
  - Id – unique identifier of the Entity (see note in Section 5.3 below)
  - Status – status of the Entity.
- Entity Link – Allows for linking and merging of entities. The link type would indicate a link or merge. Attribute descriptions:
  - entityLinkType – indicates whether the entities are linked as active “peers” or whether one is deprecated as in a merge.

Describe tie in to the RIM.

The above model provides a great deal of implementation flexibility. It is recognized that some implementations may find such flexibility a burden rather than a benefit. The two specific areas in which this may apply are in the use of different Entity Types and also the use of multiple Entity Domains.

Each individual RFP issued will need to define the scope and resulting RFP Submissions will need to take a specific stance with respect to both multiple Entity Types and Entity Domains. For those where these capabilities are not supported, the specific operations relating to them would not apply, and recognition of separate Entity Domain and Entity Type Identifiers would be unnecessary in many entity level and query operations. In XEIS implementations, support for multiple Entity Domains is necessary, although this could still be restricted to support of one “owning” Entity Domain. This effectively means that the Entity Domain and Entity Type Identifiers can be considered as optional at the implementation level (or at least conditional on the scope of the EIS). **Note that this is not directly indicated in the operation descriptions below**.

For the sake of understandability and completeness, the interfaces in this specification are defined on the basis of an XEIS that does manage multiple Entity Domains and multiple Entity Types. It is relatively easy to subsequently constrain these interfaces as appropriate to an XEIS with one owning Entity Domain or to an EIS that deals with only one entity type (e.g. by removing or restricting the relevant operations and inputs). Appendix IV contains a detailed discussion of these issues. This appendix contains a table which describes which operations and inputs are relevant to each main type of EIS. This deals with the optionality (or conditional behavior) mentioned above.

Also, it should be remembered that this is a functional specification. The operation “inputs” identified are the information that an EIS instance needs to know to carry out its task. In most cases, the technical specification will include these items as specific input parameters to operation calls. In some cases however, there may be other alternatives. With respect to Entity Domain and Entity Identifier, some alternatives are also discussed in Appendix IV.

In a few cases below, where complete operations are only applicable to XEIS implementations, this is explicitly indicated.

One additional area that needs further clarification is the use of codes, both at the metadata level for identifying data types for traits and also for value sets for coded traits.

Identification of data types in metadata will use the coding scheme defined for HL7 V3.

EIS also needs an explicit mechanism to identify value or code sets. The concepts as described in the existing HL7 CTS (Common Terminology Service) specification will be applied, so a “value set ID” will be used, which will reference one or more code sets, at least in the case of HL7 defined codes. CTS also suggests a similar mechanism for dealing with non-HL7 codes, which is relevant to this specification. An additional version Id attribute is included to allow for different versions of the value set to be used, although in the forthcoming CTS II it is planned that the value set may be inherently versioned, so this may become unnecessary at a later date.

The de-referencing mechanism is out of scope of the specification, but EIS implementations may include means to validate code values or call other services such as CTS to do so.

Note also that in this specification, “identifying information” is any set of traits that can be used singly or together to identify an entity (even approximately). An “identifier” is a data item or structure used to provide a unique identification. See also the introduction to section 5.3.

Status code set is undefined in this specification. It is explicitly stated below that these are left to the RFP Submission. The values need to be determined by the requirements for maintaining metadata. If the RIM code set values are appropriate, then these should be used.

5.1 Administration Functions

Administrative Functions: These functions are expected to be used only be systems administrators who maintain the operations of EIS instances. It is expected that further "operational" operations would be defined in the technical specification, e.g. start service, stop service, notify if service is operational, etc. These have not been defined in this functional model and it is assumed that this will be covered in the Technical RFP issuance and response.

5.1.1 Remove an Entity from EIS

Description	Allows for the "complete" removal of an Inactive Entity from an EIS service.
Precondition	The Entity should exist and should be in an Inactive State The Entity should not be linked to other Entities
Inputs	▪ Entity Domain Identifier ▪ Entity Type Identifier ▪ Entity Identifier
Outputs	An acknowledgement that the Entity has been removed
Invariants	
Postconditions	None
Exception Conditions	▪ Identifier is not recognized as identifying an entity ▪ Identifier identifies an entity that is active ▪ The Entity is linked to other Entities
Aspects left to RFP Submitters	RFP Submissions must define whether the Entity Identifier may be reused after this operation.
Relationship to levels of conformance	
Miscellaneous notes	The concept of "removal" implies that there is some persistent storage of identity information within the implementation of the service. This may be achieved in many different ways, but this is outside the scope of the interface definition. This SFM assumes that entity identifiers cannot be reused, but this is an issue that must be defined precisely in RFP submissions.
Other relevant content	

5.2 Service Metadata Management Functions

This set of functions in this set includes the management of traits, i.e. creation, retrieval, updating and deletion of traits. It is likely that instances of the system will be *bootstrapped* with a small set of default metadata information. This can be decided as part of the Technology specification and/or specific profiles.

In actual implementations, the system may generate identifiers that would be used to create and maintain relationships and allow for system administration etc. In general, these would not be displayed to the end user. These would be distinct from the "Identifier" used in many of the operations below, which is explicitly a user defined identifier (such as a mnemonic) that the users would use to identify instances of metadata classes.

Instances of EIS could also provide these capabilities using static configuration files. The interfaces are explicitly defined to enable implementations that wish to support a more dynamic structuring model to function in a consistent manner.

5.2.1 Create a Entity Domain (XEIS only)

Description	Allows for definition of a new Entity Domain. This allows for setting up Entity Domains as metadata. Entity Domains provide the capability to allow different organizational or location contexts for Entities.
Precondition	Entity Domain does not already exist on the system
Inputs	➢ Entity Domain Metadata: ➢ Identifier ➢ Name ➢ Description ➢ Status ➢ ForCrossReference
Outputs	➢ An acknowledgement that the Entity Domain has been successfully been created
Invariants	
Postconditions	Entity Domain definition created on system
Exception Conditions	➢ Entity Domain already exists on system (Identifier must be unique)
Aspects left to RFP Submitters	Values and associated meaning of Status need to be defined (e.g. pending, active, deprecated, marked for deletion) It is suggested that a value of "master" could be reserved and used in hierarchic multi Entity Domain cases.
Relationship to levels of conformance	
Miscellaneous notes	This interface is only relevant to XEIS implementations.
Other relevant content	

5.2.2 Update a Entity Domain Definition (XEIS only)

Description	Allows for update of a Entity Domain definition.
Precondition	Entity Domain already exists on the system
Inputs	➢ Entity Domain Identifier ➢ Entity Domain Metadata: (all optional) ➢ Name ➢ Description ➢ Status ➢ ForCrossReference
Outputs	An acknowledgement that the Entity Domain has been successfully been updated
Invariants	
Postconditions	Entity Domain metadata updated
Exception Conditions	➢ Entity Domain does not exist on system ➢ Identifier must be unique
Aspects left to RFP Submitters	It is recommended that status is prevented from being set to inactive if the Entity Domain is still used in Trait Assignments for active Entities. It is recommended that status is prevented from being set to inactive if the Entity Domain is still associated with active Entity instances.”
Relationship to levels of conformance	
Miscellaneous notes	This interface is only relevant to XEIS implementations.
Other relevant content	

5.2.3 Retrieve a Entity Domain Definition (XEIS only)

Description	Allows for viewing of a Entity Domain definition.
Precondition	Entity Domain already exists on the system
Inputs	➢ Entity Domain Identifier
Outputs	➢ Entity Domain Metadata: ➢ Name ➢ Description ➢ Status ➢ ForCrossReference
Invariants	
Postconditions	None
Exception Conditions	➢ Entity Domain does not exist on system
Aspects left to RFP submitters	This is an optional operation. The same functionality could be provided by “List Entity Domains” in the Query interface
Relationship to levels of conformance	
Miscellaneous notes	This interface is only relevant to XEIS implementations.
Other relevant content	

5.2.4 Create an Entity Type Classifier

Description	Allows for definition of a new Entity Type Classifier. This allows for setting up Entity Type Classifiers as metadata. Entity Type Classifiers provide the capability to define different sets or groups of Entity Types.
Precondition	Entity Type Classifier does not already exist on the system
Inputs	➢ Entity Type Classifier Identifier ➢ Entity Type Classifier Metadata: ➢ Name ➢ Description ➢ Status
Outputs	➢ An acknowledgement that the Entity Type Classifier has been successfully been created
Invariants	
Postconditions	Entity Type Classifier definition created on system
Exception Conditions	➢ Entity Type Classifier already exists on system (Identifier must be unique)
Aspects left to RFP submitters	Values and associated meaning of Status need to be defined (e.g. pending, active, deprecated, marked for deletion)
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.2.5 Update an Entity Type Classifier Definition

Description	Allows for update of a Entity Type Classifier definition.
Precondition	Entity Type Classifier already exists on the system
Inputs	➢ Entity Type Classifier Identifier ➢ Entity Type Classifier Metadata: (Optional) ➢ Name ➢ Description ➢ Status
Outputs	An acknowledgement that the Entity Type Classifier has been successfully been updated
Invariants	
Postconditions	Entity Type Classifier metadata updated
Exception Conditions	➢ Entity Type Classifier does not exist on system ➢ Identifier must be unique
Aspects left to RFP submitters	It is recommended that status is prevented from being set to inactive if the Entity Type Classifier still has active Entity Types.
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.2.6 Retrieve an Entity Type Classifier Definition (optional)

Description	Allows for viewing of an Entity Type Classifier definition.
Precondition	Entity Type Classifier already exists on the system
Inputs	➢ Entity Type Classifier Identifier
Outputs	➢ Entity Type Classifier Identifier ➢ Entity Type Classifier Metadata: ➢ Name ➢ Description ➢ Status
Invariants	
Postconditions	None
Exception Conditions	➢ Entity Type Classifier does not exist on system
Aspects left to RFP submitters	This is an optional operation. The same functionality could be provided by "List Entity Type Classifiers" in the Query interface
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.2.7 Create an Entity Type

Description	Allows for definition of a new Entity Type within a specific Entity Type Classification Scheme. Entity Types provide metadata that can then be assigned to Entity Domains and instantiated within EIS.
Precondition	Entity Type does not already exist on the system
Inputs	➢ Entity Type Identifier ➢ Entity Type Metadata: ➢ Name ➢ Description ➢ Status ➢ Entity Type Classifier Identifier ➢ List of Associated Entity Domain Identifiers
Outputs	➢ An acknowledgement that the Entity Type has been successfully been created
Invariants	
Postconditions	Entity Type definition created on system
Exception Conditions	➢ Entity Type already exists on system (Identifier must be unique) ➢ Entity Type Classifier does not exist on system ➢ Entity Domain does not exist on system
Aspects left to RFP submitters	Values and associated meaning of Status need to be defined (e.g. pending, active, deprecated, marked for deletion)
Relationship to levels of conformance	
Other relevant content	

5.2.8 Update an Entity Type Definition

Description	Allows for update of an Entity Type definition. Includes association and disassociation with Entity Domains
Precondition	Entity Type already exists on the system

Inputs	➢ Entity Type Identifier
	➢ Entity Type Metadata: (Optional)
	  ➢ Name
	  ➢ Description
	  ➢ Status
	  ➢ List of Associated Entity Domain Identifiers

Outputs	An acknowledgement that the Entity Type has been successfully been updated
Invariants	
Postconditions	Entity Type metadata updated
Exception Conditions	➢ Entity Type does not exist on system ➢ Identifier must be unique ➢ Entity Domain does not exist on system
Aspects left to RFP submitters	Association and removal from Entity Domains could be treated as a separate operation, but seemed to be more appropriate as part of the Entity Type metadata update. It is recommended that status is prevented from being set to inactive if any Entities of the type are active Also, could include assignment and de-assignment of Traits here to keep things more coarse grained.
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.2.9 Retrieve an Entity Type Definition (optional)

Description	Allows for viewing of an Entity Type definition.
Precondition	Entity Type already exists on the system
Inputs	➢ Entity Type Identifier
Outputs	➢ Entity Type Metadata: ➢ Name ➢ Description ➢ Status ➢ List of Associated Entity Domain Identifiers
Invariants	
Postconditions	None
Exception Conditions	➢ Entity Type does not exist on system
Aspects left to RFP submitters	This is an optional operation. The same functionality could be provided by "List Entity Types" in the Query interface
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.2.10 Create a Trait Definition

Description	Allows for definition of a new Trait type that may be assigned to Entity Types and then used to identify Entities of that Type.
Precondition	Trait does not already exist on the system
Inputs	➢ Trait Identifier ➢ Trait Metadata: ➢ Name ➢ Data Type ➢ Value Set ID (optional – mandatory if data type is a code) ➢ Version ID (optional – mandatory if data type is a code) ➢ Description ➢ Status
Outputs	➢ An acknowledgement that the trait has been successfully been created
Invariants	
Postconditions	Trait definition created on system
Exception Conditions	➢ Trait already exists on system (Identifier must be unique) ➢ Data Type not recognized
Aspects left to RFP submitters	Values and associated meaning of Status need to be defined (e.g. pending, active, deprecated, marked for deletion)
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.2.11 Update a Trait Definition

Description	Allows for update of a Trait type definition.
Precondition	Trait already exists on the system
Inputs	➢ Trait Identifier ➢ Trait Metadata: (Optional) ➢ Name ➢ Data Type ➢ Value Set ID (optional – mandatory if data type is a code) ➢ Version ID (optional – mandatory if data type is a code) ➢ Description ➢ Status
Outputs	An acknowledgement that the trait has been successfully been updated
Invariants	
Postconditions	Trait metadata updated
Exception Conditions	➢ Trait does not exist on system ➢ Identifier must be unique ➢ Data Type not recognized
Aspects left to RFP submitters	It is recommended that status is prevented from being set to inactive if active trait assignments.
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.2.12 Retrieve Trait Definition (optional)

Description	Allows for viewing of a Trait type definition.
Precondition	Selected Trait already exists on the system
Inputs	➢ Trait Identifier
Outputs	➢ List of Trait Metadata ➢ Name ➢ Data Type ➢ Value Set ID ➢ Version ID ➢ Description ➢ Status
Invariants	
Postconditions	None
Exception Conditions	➢ Trait does not exist on system
Aspects left to RFP submitters	This is an optional operation. The same functionality could be provided by “List Traits” in the Query interface
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.2.13 Assign Trait To Entity Type

Description	Allows for assignment of a trait to an entity type within a specific domain or domains. Trait can be set as mandatory or not for entities of the Entity Type.
Precondition	The Entity Type specified by the Entity Type Identifier and the Trait specified by the Trait Identifier exist. Entity Type is associated with specified Entity Domains. The Trait is not already associated with the Entity Type and Entity Domains specified.
Inputs	➢ Trait Identifier ➢ Entity Type Identifier ➢ List of Entity Domain Identifiers ➢ Mandatory indicator (Y/N) for each
Outputs	An acknowledgement that the trait has been successfully been assigned with the entity type
Invariants	
Postconditions	Trait assigned to Entity Type for specified Entity Domains
Exception Conditions	➢ Entity Type does not exist in the system ➢ Entity Domain does not exist in the system ➢ Entity Type not associated with specified Entity Domain ➢ Trait does not exist in the system
Aspects left to RFP submitters	It may make sense to allow for a number of traits to be assigned in a single operation, but since this is an admin function which would not be used frequently this has not been specified. Presumably, only Traits with appropriate Status should be able to be assigned (e.g. “active”)
Relationship to levels of conformance	
Miscellaneous notes	Where Entity Type and/or Entity Domain are fixed or not supported, the mandatory (Y/N) indicator is still needed and would apply to the single default entity type and/or domain
Other relevant content	

5.2.14 Un-assign Trait from Entity Type

Description	Allows for “un-assignment” of a trait from an entity type within a specific Entity Domain or Domains
Precondition	The Entity Type specified by the Entity Identifier and the Trait specified by the Trait Identifier both exist. The Trait is assigned to the Entity Type and Entity Domains specified.
Inputs	➢ Trait Identifier ➢ Entity Type Identifier ➢ List of Entity Domain Identifiers
Outputs	An acknowledgement that the trait has been successfully been unassigned from the Entity Type
Invariants	
Postconditions	Trait no longer assigned to Entity Type for the specified Entity Domains
Exception Conditions	➢ Entity Type does not exist in the system ➢ Entity Domain does not exist in the system ➢ Trait does not exist in the system ➢ Trait not assigned to the Entity Type and Entity Domain
Aspects left to RFP submitters	Need to decide whether to give a warning if entities exist with the value of the trait set.
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.2.15 Update Trait Assignment to Entity Type

Description	Allows for update of an assignment between a trait and an entity type within a specific Entity Domain or Domains
Precondition	The Entity Type specified by the Entity Identifier and the Trait specified by the Trait Identifier both exist. The Trait is assigned to the Entity Type and Entity Domains specified.
Inputs	➢ Trait Identifier ➢ List of Entity Domain Identifiers ➢ Entity Type Identifier ➢ Mandatory Indicator
Outputs	An acknowledgement that the trait assignment has been successfully been updated
Invariants	
Postconditions	Trait assignment updated
Exception Conditions	➢ Entity Type does not exist in the system ➢ Entity Domain does not exist in the system ➢ Trait does not exist in the system ➢ Trait not assigned to the Entity Type and Entity Domain
Aspects left to RFP submitters	This operation is optional. Only foreseen update is to change whether or not the trait is mandatory. This could be bundled into the Entity Type update instead of a separation operation.
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.3 Entity Management Functions

Note: The use of a unique user definable entity identifier has been treated as optional in this specification. This would be in addition to any identifying traits. It is left to the RFP Submitter whether this should remain, or whether to allow this to be dependent on the entity type. The Create an Entity operation description below assumes it is optional, however several following update and retrieval operations take the identifier as mandatory input. The RFP submitter may choose to make this mandatory or find another mechanism, such as use of the assumed system generated (hidden) identifier to achieve this purpose, which may require exposing the hidden identifier in some cases. In the case of the "unattended encounter" use case, generation of an identifier is really necessary. Where manual input of the identifier is permitted, the implementation must provide duplicate checking (as defined in 5.3.1 below).

5.3.1 Create an Entity

Description	Allows for creation of an entity with a list of trait values.
Precondition	The traits listed in the input are defined in the metadata for the Entity Type.
Inputs	➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Entity Identifier (optional) ➢ Entity Status ➢ Trait List (name, value pairs): must include any traits identified as mandatory (for the entity type)
Outputs	An acknowledgement that the Entity has been successfully created. Return generated Entity Identifier if this is generated.
Invariants	
Postconditions	An Entity is created
Exception Conditions	➢ Entity already exists within EIS (Identifier must be unique for the Entity Type / Entity Domain combination) ➢ A trait in the list does not exist in the system metadata for the Entity Type ➢ There are duplicates in the trait list ➢ Insufficient Traits: Values of all mandatory traits for the Entity Type have not been provided

	➢ Invalid data type or code value for trait
Aspects left to RFP submitters	Where identifier is not being input, some means is necessary to identify potential duplicates. This may necessitate matching on other key mandatory traits.
Relationship to levels of conformance	Included in Functional profile: Entity Update
Miscellaneous notes	Although it does not directly impact the description of the interface or parameters, it is expected that EIS implementations will typically provide some level of automated implicit linking capabilities. This could be policy driven, or be handled manually using the explicit linking operations described in this specification. This functionality is similar to that described by the IHE PIX profile. This would be triggered when a new entity is created or the traits of an entity updated. The triggered behavior should be as described for the Link Entity operation. Actually “merging” entities based on automated logic is not encouraged. Note that the actual policies are handled through “out of band” agreements.
Other relevant content	

5.3.2 Inactivate an Entity

Description	Makes an Entity inactive, i.e. will not appear in search result sets.
Precondition	The Identifier identifies an Entity that is active
Inputs	➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Entity Identifier
Outputs	Confirmation that the Entity identified by the Identifier is now inactive
Invariants	Entity is unchanged except for active/inactive status
Postconditions	Entity status is set to inactive
Exception Conditions	➢ Identifier is not recognized as identifying an Entity ➢ Identifier identifies an Entity that is inactive
Aspects left to RFP submitters	A suggestion was made that instead of having precondition of Entity being active, we could allow inactive Entities, and the function then must ensure that the Entity is inactive, irrespective of previous state. Description would then read "Ensure Entity is inactive". Preference is expressed to leave it as described, since it seems more robust to reject this as an error, since it may be inactivating the wrong Entity.
Relationship to levels of conformance	Included in Functional profile: Entity Update
Miscellaneous notes	
Other relevant content	

5.3.3 Activate an Entity

Description	Changes an Inactive Entity to an Active state
Precondition	The Entity should exist and should be in an Inactive State
Inputs	➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Entity Identifier
Outputs	Confirmation that the Entity identified by the Identifier is now active
Invariants	Entity is unchanged except for active/inactive status
Postconditions	Entity Status is active
Exception Conditions	➢ Identifier is not recognized as identifying an entity ➢ Identifier identifies an entity that is active
Aspects left to RFP submitters	
Relationship to levels of conformance	Included in Functional profile: Entity Update
Miscellaneous notes	
Other relevant content	

5.3.4 Merge Entities

Description	Allows for the merging of two entities. The Target Entity Identifer is the "winner" in the merging operation. Deprecated Entity may be automatically set to Inactive status. Merge operation assumed to be within a single Entity Domain and Entity Type only. Use linking otherwise.
Precondition	Both Entities to be merged should exist in the system. Target Entity must be in Active state. Both Entities must be same Entity Domain and Entity Type.
Inputs	➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Target Entity Identifier ➢ Entity Identifier to be merged (deprecated) ➢ Inactivation Requested (whether deprecated entity should be inactivated)
Outputs	An acknowledgement that the Entities have been merged.
Invariants	
Postconditions	Deprecated Entity set to Inactive state if requested
Exception Conditions	➢ Target Entity does not exist ➢ Merged Entity does not exist
Aspects left to RFP submitters	
Relationship to levels of conformance	Included in Functional profile: Entity Merge
Miscellaneous notes	If automated implicit linking capabilities are provided, then the rules for dealing with such links need to be specified in implementations.
Other relevant content	

5.3.5 Undo Merge Entities

Description	Allows for the unmerging of exactly two entities for searches. This is a specific "reversal" or contra, which would not therefore require manual intervention to say which properties should go where. Would need access to original inactivated Entity details.
Precondition	The Entity resulted from a previous merge, and no changes have been applied to the Entity since the merge operation.
Inputs	➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Entity Identifier-1 ➢ Entity Identifier-2
Outputs	➢ An acknowledgement that the Entities have been unmerged ➢ Unmerged (reinstated) Entity Identifier
Invariants	
Postconditions	Both entities returned to their previous state before the Merge operation.
Exception Conditions	➢ Identifier is not recognized as identifying an entity ➢ No additional merged Entity exists for this Entity ➢ Updates have occurred since merge that renders undo merge un-deterministic
Aspects left to RFP submitters	Rather than specifically requiring the Entity Identifier of the deprecated Entity, the implementation may offer an operation that shows recent merge transactions related to one Entity and reverse the transaction.
Relationship to levels of conformance	Included in Functional profile: Entity Merge
Miscellaneous notes	
Other relevant content	

5.3.6 Unmerge Entities

Description	Allows for the unmerging of two Entities. This would not carry out an "undo" but reinstate a previously deprecated Entity. Note that the state of the unmerged Entities cannot be guaranteed to be valid and assumes that some subsequent manual updates would be carried out to correct both entries.
Precondition	The Entity resulted from a previous merge.
Inputs	➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Entity Identifier-1 ➢ Entity Identifier-2
Outputs	➢ An acknowledgement that the Entities have been unmerged ➢ Unmerged (reinstated) Entity Identifier
Invariants	
Postconditions	(end up with two duplicate records requiring further work)
Exception Conditions	➢ Identifier is not recognized as identifying an entity ➢ No additional merged Entity exists for this Entity
Aspects left to RFP submitters	From an overall process perspective, the “unmerge” should require manual intervention afterwards. The RFP submitter may choose to enforce an inactive status on the unmerged entities until a further update is carried out.
Relationship to levels of conformance	Included in Functional profile: Entity Merge
Miscellaneous notes	
Other relevant content	

5.3.7 Link Entities

Description	Allows for the linking of two entities. Linking operation can be carried out within a single domain or across two different domains.
Precondition	Both Entities to be linked should exist in the system. Target Entity must be in Active state.
Inputs	➢ Entity Domain Identifier 1 ➢ Entity Type Identifier 1 ➢ Entity Identifier 1 ➢ Entity Domain Identifier 2 ➢ Entity Type Identifier 2 ➢ Entity Identifier 2
Outputs	An acknowledgement that the Entities have been linked.
Invariants	
Postconditions	Both Entities updated with link information.
Exception Conditions	➢ Entity 1 does not exist ➢ Entity 2 does not exist ➢ Entities already linked
Aspects left to RFP submitters	For cross-Entity-Domain linking, RFP Submitters may choose to allow “peer to peer” linking or enforce that the link is to and from a “Master” Entity Domain for the XEIS instance. Note that this does not affect the structure of the interface, but is an important behavioral aspect. These interfaces currently permit linking of entities of different types, e.g. from a Person to a Patient. Submitters need to consider whether to restrict this to linking entities of the same type only.
Relationship to levels of conformance	Included in Functional profile: Entity Cross-Reference
Miscellaneous notes	Note – this operation could be triggered by implicit logic as part of the “create entity” or “update entity traits” operations.
Other relevant content	

5.3.8 Unlink Entities

Description	Allows for the unlinking of two entities.
Precondition	Both Entities to be unlinked should exist in the system and be linked to each other.
Inputs	➢ Entity Domain Identifier 1 ➢ Entity Type Identifier 1 ➢ Entity Identifier 1 ➢ Entity Domain Identifier 2 ➢ Entity Type Identifier 2 ➢ Entity Identifier 2
Outputs	➢ An acknowledgement that the Entities have been unlinked.
Invariants	
Postconditions	Both Entities updated with unlink information.
Exception Conditions	➢ Entity 1 does not exist ➢ Entity 2 does not exist ➢ Entities not linked
Aspects left to RFP submitters	
Relationship to levels of conformance	Included in Functional profile: Entity Cross-Reference
Miscellaneous notes	
Other relevant content	

5.3.9 Update Entity Trait Values

Description	Allows for addition and/or update of a set of trait values for an entity specified by a unique Entity Identifier
Precondition	The Entity specified by the Entity Identifier exists.
Inputs	➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Entity Identifier ➢ List (Trait name, value pairs)
Outputs	An acknowledgement that the trait values have been successfully added and/or updated
Invariants	
Postconditions	Entity updated with trait values
Exception Conditions	➢ Entity does not exist in the system ➢ A trait in the list does not exist in the system metadata for the Entity Type ➢ There are duplicates in the trait list ➢ Insufficient Traits: Value of a mandatory trait has been set to null ➢ Invalid data type or code valaue for trait
Aspects left to RFP submitters	
Relationship to levels of conformance	Included in Functional profile: Entity Update
Miscellaneous notes	Although it does not directly impact the description of the interface or parameters, it is expected that EIS implementations will typically provide some level of automated implicit linking capabilities. This could be policy driven, or be handled manually using the explicit linking operations described in this specification. This functionality is similar to that described by the IHE PIX profile. This would be triggered when a new entity is created or the traits of an entity updated. The triggered behavior should be as described for the Link Entity operation. Actually "merging" entities based on automated logic is not encouraged. Note that the actual policies are handled

	through “out of band” agreements.
Other relevant content	

5.4 Query Functions

5.4.1 Get All Information for an Entity

Description	Retrieves all information for an Entity known by the EIS (traits, status etc)
Precondition	The Identifier identifies an Entity that exists
Inputs	➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Entity Identifier
Outputs	Status and all trait values of the Entity identified by the Identifier If entity has been merged into another "main" identifier, then return active identifier.
Invariants	Entity is unchanged.
Postconditions	N/A
Exception Conditions	➢ Identifier is not recognized as identifying an entity
Aspects left to RFP submitters	N/A
Relationship to levels of conformance	Included in Functional profile: Entity Query Minimum sets of Traits for an Entity Type will be defined in semantic profiles.
Miscellaneous notes	
Other relevant content	

5.4.2 Find Entities by Trait

Description	Given a list of Traits (Trait Identifier or Name, Trait Value pairs), this allows for a search of matching Entities. Outputs include a quality of match.
Precondition	The Traits specified in the list exist in the system
Inputs	➢ A list of (Trait Name or Trait Identifier, Trait Value) pairs ➢ Requested confidence of match ➢ Trait Identifiers or Names to be returned (can be different than input -default) ➢ Matching algorithm (Optional) ➢ List of Entity Type Identifiers (Optional) ➢ List of Entity Domain Identifiers (Optional)
Outputs	➢ An acknowledgement that a list of Entities for the search predicate has been found or no matches. ➢ A list of (Entity Type Identifier, Entity Identifier, quality_of_match) matching the search predicate, in order of the quality_of_match, together with requested output Traits.
Invariants	
Postconditions	None
Exception Conditions	➢ A Trait Identifier/Name in the input list does not exist ➢ Entity Type does not exist ➢ Entity Domain does not exist
Aspects left to RFP submitters	It is left to OMG RFP submitters whether to define specific queries for specific entity types and trait sets and/or algorithms etc. For XEIS instances, may also be useful to return a list of Entity Domains that were covered and/or not covered by the search. A means needs to be provided for authorized roles to retrieve entities that have an “inactive” status. This may be provided by including status as an input or other mechanism. Default behavior of this operation should return “active” matches.
Relationship to levels of conformance	Included in Functional profile: Entity Query

Miscellaneous notes	The confidence and quality of match inputs and outputs should provide the ability to define a two-tiered match threshold criteria, such as "definite match." not requiring human intervention, and "presumptive match," requiring further verification. The response quality could be used to determine which of the thresholds were achieved. Note that these values can be composite data structures, numerical, ontological, etc.

5.4.3 List Linked Entities

Description	Given an Entity Identifier, list all other entities that are linked to the Entity (optionally constrained within one or more Entity Domains)
Precondition	The Entity exists in the specified domain.
Inputs	➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Entity Identifier ➢ List of other Entity Domain Identifiers (optional)
Outputs	List of Entities (Entity Domain Identifier, Entity Type Identifier, Entity Identifier) that are linked to the specified Entity
Invariants	
Postconditions	None
Exception Conditions	➢ Entity Domain not known to the system ➢ Entity not known to the system
Aspects left to RFP submitters	
Relationship to levels of conformance	Included in Functional profile: Entity Cross-Reference
Miscellaneous notes	
Other relevant content	

5.4.4 Request Entity Update Notifications

Description	The service consumer lodges a request to be notified if the EIS becomes aware of any changes to information for a specific Entity (traits, status or entity links) or entities of a specific type and/or domain combination.
Precondition	Where specified, the Entity exists in the specified domain. Where entered, Entity Domain and Entity Types are supported.
Inputs	1. Subscriber identification and/or destination One or more of: ➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Entity Identifier
Outputs	Acknowledgement of request. (Potentially a series of asynchronous update notifications)
Invariants	
Postconditions	None
Exception Conditions	➢ Entity Domain not known to the system ➢ Entity Type not known to the system ➢ Entity not known to the system
Aspects left to RFP submitters	
Relationship to levels of conformance	Included in Functional profile: Entity Update Notification
Miscellaneous notes	This is effectively a subscription operation. This could be implemented using a publish-and-subscribe capability.
Other relevant content	

5.4.5 Notify Entity Updates

Description	The service produces a notification of an update that has been made to any information relating to a specific entity or entities within a Entity Domain and/or Entity Type (previously notified by a “Request Update Notifications” request.
Precondition	The EIS has become aware of an update to information relating to a specific Entity. A previous “Request Update Notifications” request has been received.
Inputs	➢ None
Outputs	➢ Identifier, status and all trait values of the Entity
Invariants	
Postconditions	None
Exception Conditions	
Aspects left to RFP submitters	
Relationship to levels of conformance	Included in Functional profile: Entity Update Notification
Miscellaneous notes	This is effectively a publication operation resulting from an earlier subscription operation. This could be implemented using a publish-and-subscribe capability.
Other relevant content	

5.4.6 List Entities by Entity Type and Domain

Description	Given an Entity Type Identifier, Entity Domain Identifier or both, list all matching entities
Precondition	The Entity Types and Entity Domains entered exist.
Inputs	➢ Entity Domain Identifier AND/OR Entity Type Identifier ➢ List of Traits (Identifiers or Names) (Optional) ➢ Entity Status (Optional)
Outputs	List of Entities that match the entered Entity Type and/or Entity Domain and status if entered combination. For each entity, return: ➢ Entity Domain Identifier ➢ Entity Type Identifier ➢ Identifier ➢ Status ➢ List (Trait name, value pairs) for requested traits
Invariants	
Postconditions	None
Exception Conditions	➢ Entity Domain not known to the system ➢ Entity Type not known to the system
Aspects left to RFP submitters	
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.4.7 List Supported Entity Classifiers and Entity Types

Description	Provides a list of Entity Type Classifiers and Entity Types supported by the EIS and the Entity Domains with which each is associated. This can be all Entity Type Classifiers or for one or more specific named Entity Type Classifiers only.
Precondition	Entity Type Classifiers specified already exist on the system
Inputs	➢ List of Entity Type Classifier Identifiers or Names (Optional)
Outputs	➢ List of Entity Type Classifiers ➢ Name ➢ Description ➢ Status ➢ List of Entity Types ➢ Identifier ➢ Name ➢ Description ➢ Status ➢ List of associated Entity Domain Identifiers and Names
Invariants	
Postconditions	None
Exception Conditions	➢ Entity Type Classifier does not exist on system
Aspects left to RFP submitters	
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.4.8 List Supported Entity Domains (XEIS only)

Description	Provides a list of Entity Domains supported by the EIS and Entity Types with which each are associated.
Precondition	None
Inputs	➢ None
Outputs	➢ List of Entity Domains ➢ Identifier ➢ Name ➢ Description ➢ Status ➢ List of Associated Entity Types (Classifier Identifier and Name, Entity Type Identifier and Name)
Invariants	
Postconditions	None
Exception Conditions	
Aspects left to RFP submitters	
Relationship to levels of conformance	
Miscellaneous notes	This operation is only relevant for XEIS implementations.
Other relevant content	

5.4.9 List Trait Definitions

Description	Allows for viewing of Trait type definitions, either all or one or more selected Identifiers. Includes returning currently assigned Entity Types and Entity Domains.
Precondition	Selected Traits already exists on the system
Inputs	➢ List of Trait Identifiers or Trait Names (Optional)
Outputs	➢ List of Traits Metadata ➢ Identifier ➢ Name ➢ Data Type ➢ Value Set ID ➢ Version ID ➢ Description ➢ Status ➢ Assigned Entity Type and Entity Domain identifiers and Names
Invariants	
Postconditions	None
Exception Conditions	➢ Trait does not exist on system
Aspects left to RFP submitters	
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	

5.4.10 List Supported Conformance Profiles

Description	Produces a list of the Conformance Profiles that are supported by an EIS instance. (see Section 6 for a definition of Conformance Profile).
Precondition	EIS instance knows which Conformance Profiles it supports (e.g. set by configuration)
Inputs	None
Outputs	➢ List of Conformance Profiles ➢ Registering Authority Identifier ➢ Identifier ➢ Version ➢ Name ➢ Description
Invariants	
Postconditions	➢ A list of the conformance profiles supported has been delivered to the requestor
Exception Conditions	➢ Conformance profiles are not available
Aspects left to RFP submitters	Means to define and maintain which conformance profiles are supported by an EIS instance. Describing conformance assertions. Testing conformance assertions.
Relationship to levels of conformance	
Miscellaneous notes	A code system will be needed, either globally or within each context or domain which identifies conformance profiles.
Other relevant content	

6 Profiles

6.1 Introduction

A set of profiles may be defined that cover specific functions, semantic information and overall conformance. The SDF explains in detail the meaning of each of these types of profile. In brief, they are as follows:

- Functional Profile: a named list of a subset of the operations defined within this specification which must be supported in order to claim conformance to the profile.
- Semantic Profile: identification of a named set of information descriptions (semantic signifiers) that are supported by one or more operations. In the case of the EIS specification, this must identify both the Entity itself (e.g. Patient) and the Model or Classification Scheme from which it is taken (e.g. HL7 V3 RIM V2-14). This will formally define the meaning of entities and the sets of traits. In the case of HL7 based profiles, this will provide cross-references to the appropriate RIM-based domain models for the entity and each trait.
- Conformance Profile: this is a combination of a set of functional and semantic profiles taken together to give a complete coherent set of capabilities against which conformance can be claimed. Should be versioned.

Due to the fact that this is a very generic functional capability, none of the specific functional profiles are mandatory, e.g. an EIS instance could support only the Identity Cross Reference profile (similar to part of IHE's PIX) or only the Entity Identifier Update profile. However, any instance of EIS must support at least one of the functional profiles defined below. The metadata operations have not been defined in any specific profile since these may be supported in different ways within the technical specification. For a detailed discussion of the relationship to IHE profiles, see Appendix II.

No specific normative Conformance Profiles have been defined in this specification.

6.1.1 Functional Profiles:

- Entity Cross-Reference v1.0 - Includes the following operations from above:
  - Link Entities
  - Unlink Entities
  - List Linked Entities
- Entity Update v1.0 – Includes:
  - Create Entity
  - Update Entity Trait Values
  - Activate Entity
  - Inactivate Entity
- Entity Merge v1.0 – Includes:
  - Merge Entity

  - Unmerge Entity
  - Undo Merge Entity
- Entity Query v1.0 – Includes:
  - Get All Information for Entity
  - Find Entities by Trait
- Entity Update Notification v1.0 – Includes:
  - Request Entity Update Notifications
  - Notify Entity Updates

6.1.2 Semantic Profiles

This version of the specification normatively defines only one specific semantic profile, i.e. for Patient (as defined in the HL7 RIM/DIMs). This profile is stated as **optional** since a plain “vanilla” implementation of EIS that allows all metadata as input would still be of value, and Person models defined by other standard bodies would also be perfectly reasonable.

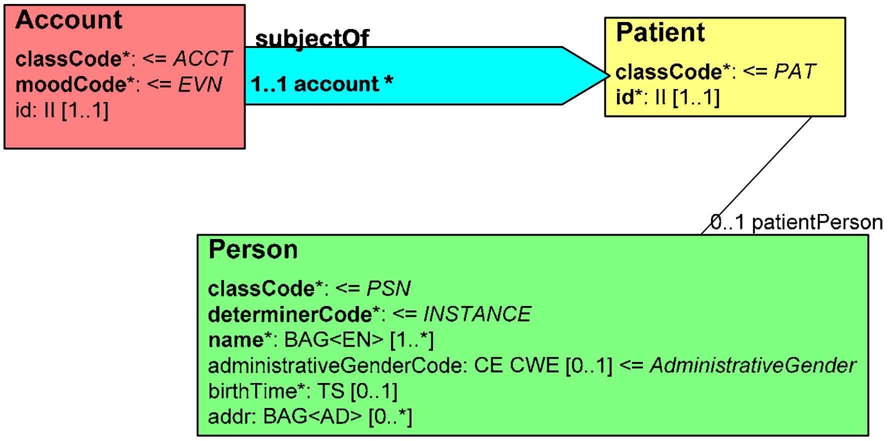
The **HL7-V3-Patient** v1.0 profile consists of the following (note that the RFP Submitter may change the actual defined metadata values as appropriate):

- Entity Type Classifier:
  a. Identifier: “HL7-V3-RIM-V2-14-Role”
  b. Name: “HL7 V3 RIM V2-14 Role sub types”
  c. Description: “This classifier identifiers entity types that are specializations of the HL7 V3 RIM V2-14 ‘Role’ class.”
  d. Status: “Active”
- Entity Type:
  a. Identifier: “HL7-V3- RIM-V2-14-Patient”
  b. Name: “HL7 V3 RIM V2-14 Patient entity”
  c. Description: “This Entity Type is the ‘Patient’ class from the HL7 V3 RIM V2-14”
  d. Status: “Active”
- Minimum Trait Set for HL7-V3-RIM-V2-14-Patient

The following minimum trait set was derived from current IHE profiles (although V3 model elements have been named as opposed to the V2 profiles used as a basis).

➢ Patient Identifier	(patient.id)
➢ Patient Name	(person.name)
➢ Date of Birth	(person.birthTime)
➢ Administrative Sex	(person.administrativeGenderCode)

➢ Patient Address	(person.addr)
➢ Patient Account Number	(account.id)



The above is a subset extracted from existing HL7 V3 domain models. Explicitly, all the content shown can be see in the Patient Billing Account RMIM (FIAB_DM000000UV01).

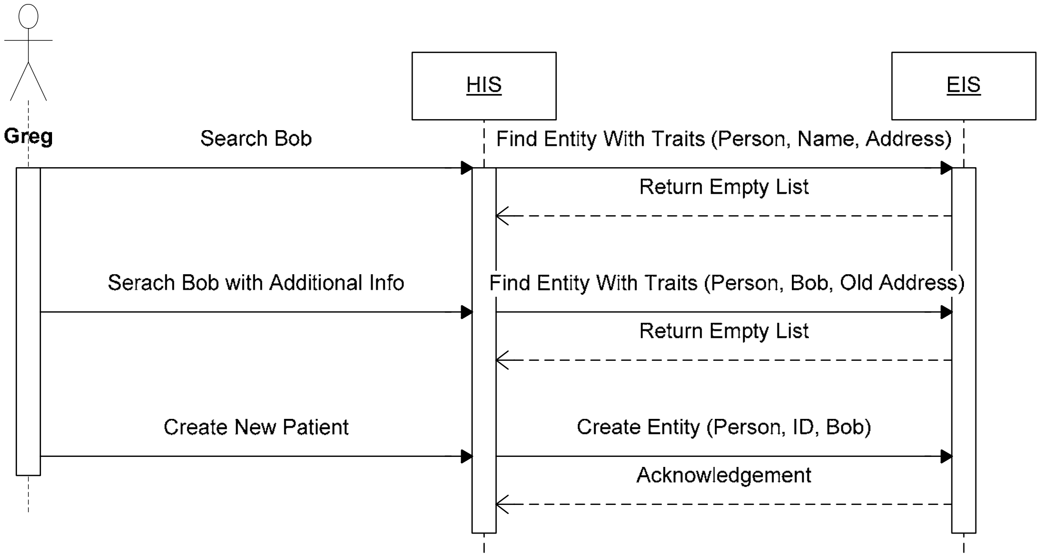
Data types and code sets are as defined in existing HL7 V3 domain models for HL7 V3 based profiles.

Non-normative suggestion - In addition to the minimum trait set, other traits can be added. It is suggested that a semantic profile is also defined with a larger set, including optional phone numbers, additional account numbers, nicknames, aliases, prior names and addresses.

7 User Scenario Interaction Details

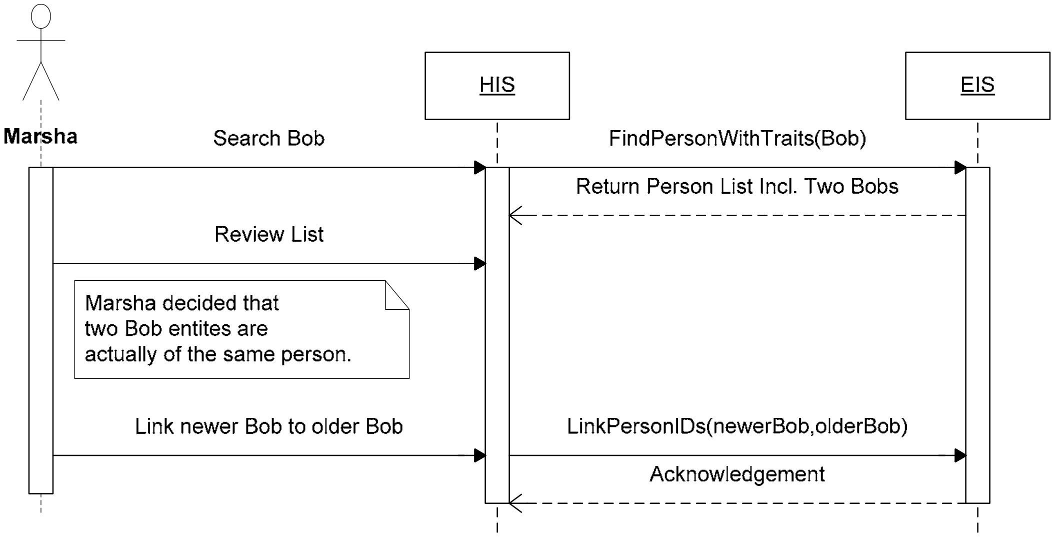
This section relates the business scenarios described earlier to some of the operations listed above. Only the main Entity Management and Query operations are covered in these scenarios. All references to EIS could be EIS or XEIS, although 7.1 through 7.8 were defined as single domain business scenarios in section 3 so would be EIS instances. Also note that in many cases, the EIS may be accessed directly from a Web or other UI application or via another system such as the HIS. In the diagrams below, most interactions are shown via the HIS, but that would be a matter for the implementation design and is not directly relevant to this specification.

7.1 Create a new patient

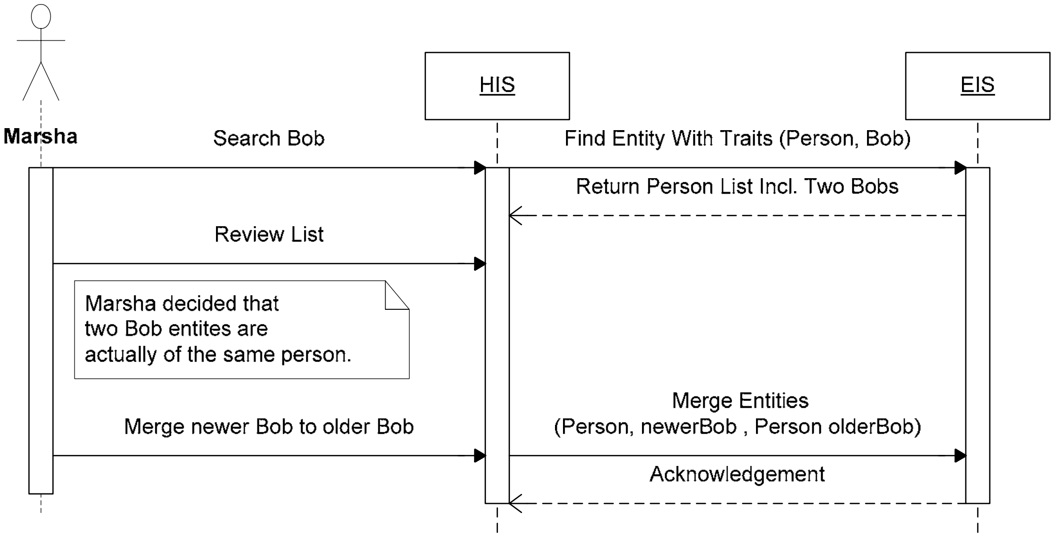

Greg
HIS
EIS
Search Bob
Find Entity With Traits (Person, Name, Address)
Return Empty List
Serach Bob with Additional Info
Find Entity With Traits (Person, Bob, Old Address)
Return Empty List
Create New Patient
Create Entity (Person, ID, Bob)
Acknowledgement

7.2 Link/Merge entities

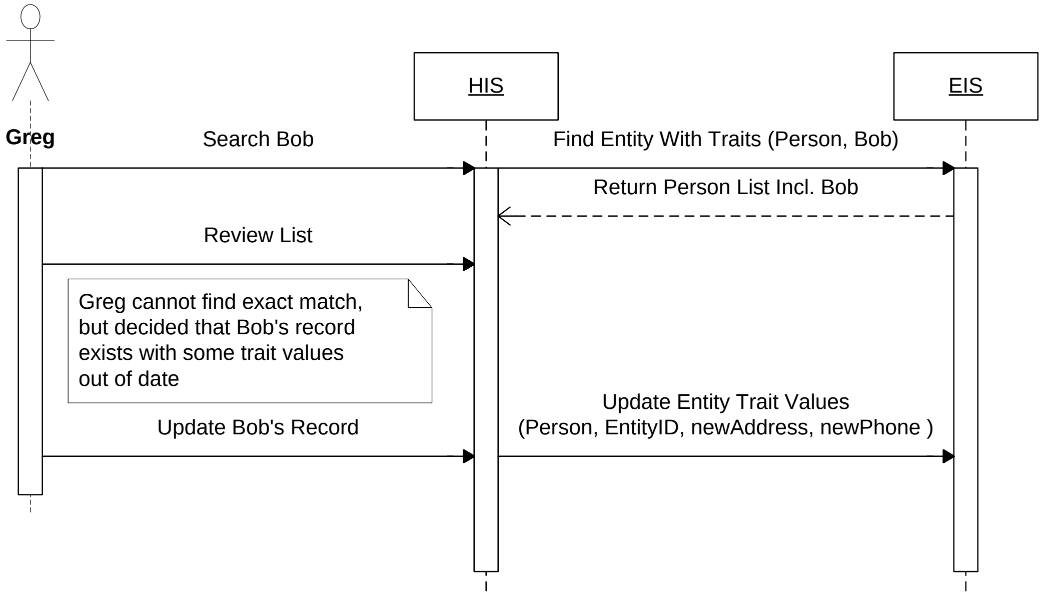
7.2.1 Link entities


Marsha
HIS
EIS
Search Bob
FindPersonWithTraits(Bob)
Return Person List Incl. Two Bobs
Review List
Marsha decided that two Bob entites are actually of the same person.
Link newer Bob to older Bob
LinkPersonIDs(newerBob,olderBob)
Acknowledgement

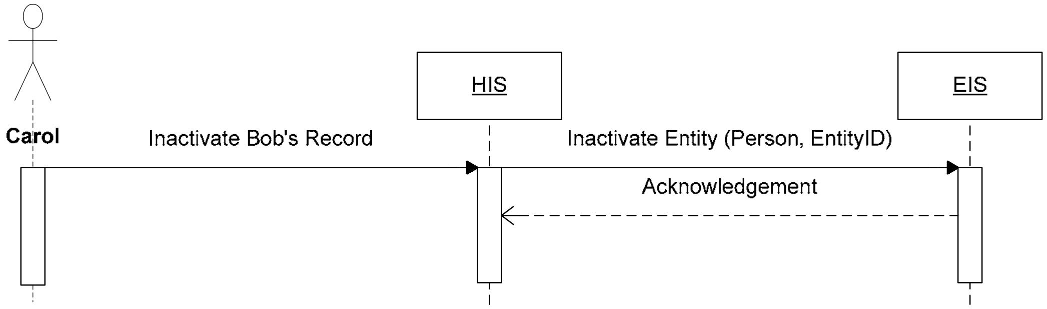
7.2.2 Merge Entities


Marsha
HIS
EIS
Search Bob
Find Entity With Traits (Person, Bob)
Return Person List Incl. Two Bobs
Review List
Marsha decided that two Bob entites are actually of the same person.
Merge newer Bob to older Bob
Merge Entities (Person, newerBob , Person olderBob)
Acknowledgement

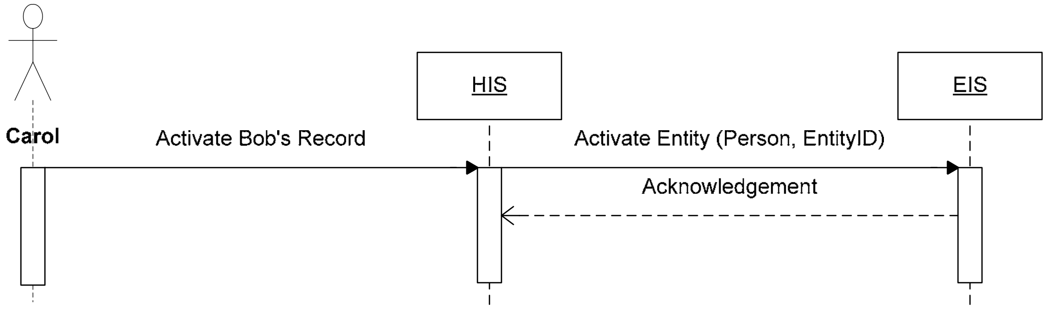
7.3 Update demographics


HIS
EIS
Greg
Search Bob
Find Entity With Traits (Person, Bob)
Return Person List Incl. Bob
Review List
Greg cannot find exact match, but decided that Bob's record exists with some trait values out of date
Update Bob's Record
Update Entity Trait Values (Person, EntityID, newAddress, newPhone)

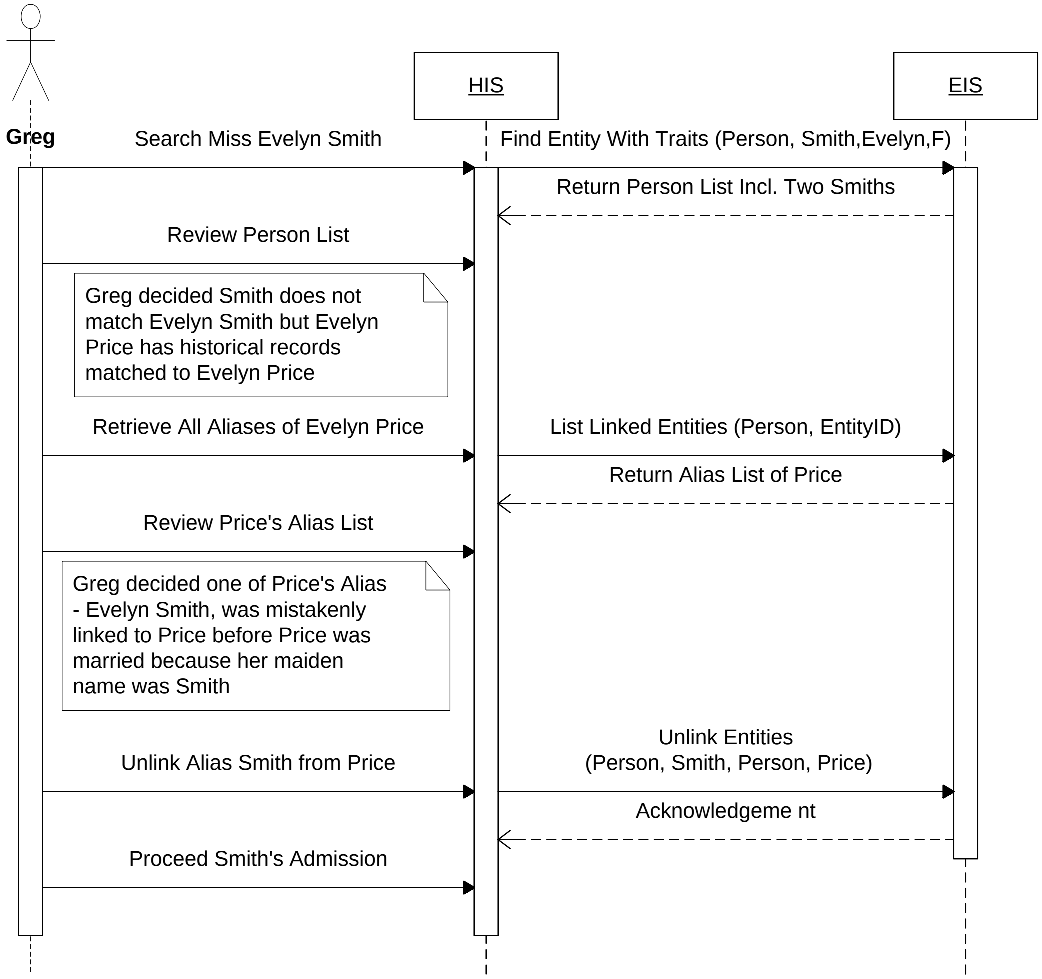
7.4 Inactivate entity from general searches



7.5 Activate (inactive) entity to general searches

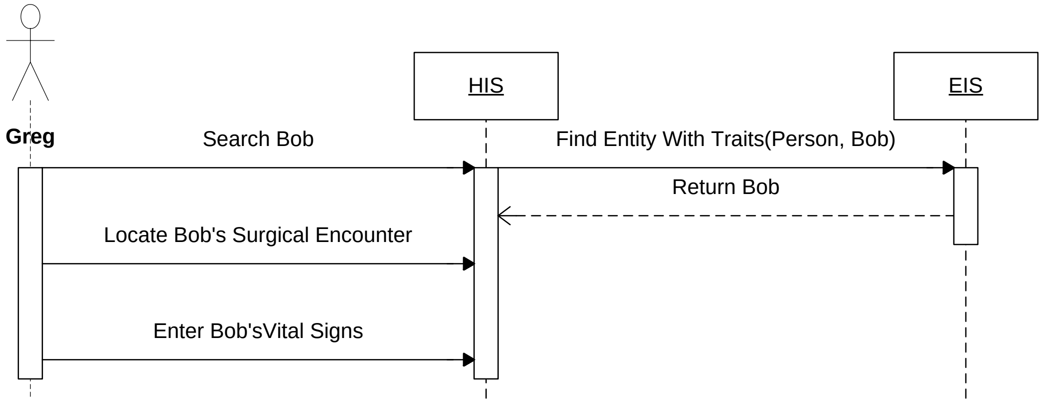


7.6 Unlink entity

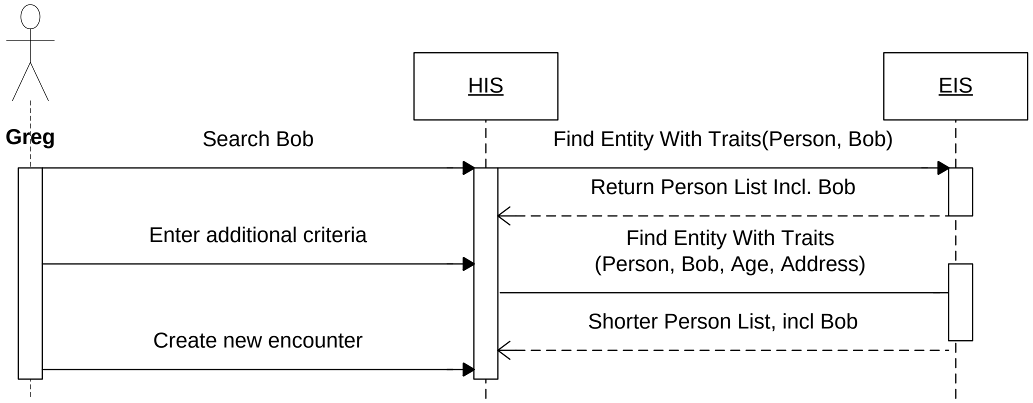

HIS
EIS
Greg
Search Miss Evelyn Smith
Find Entity With Traits (Person, Smith,Evelyn,F)
Return Person List Incl. Two Smiths
Review Person List
Greg decided Smith does not match Evelyn Smith but Evelyn Price has historical records matched to Evelyn Price
Retrieve All Aliases of Evelyn Price
List Linked Entities (Person, EntityID)
Return Alias List of Price
Review Price's Alias List
Greg decided one of Price's Alias - Evelyn Smith, was mistakenly linked to Price before Price was married because her maiden name was Smith
Unlink Alias Smith from Price
Unlink Entities (Person, Smith, Person, Price)
Acknowledgeme nt
Proceed Smith's Admission

7.7 Look up a patient

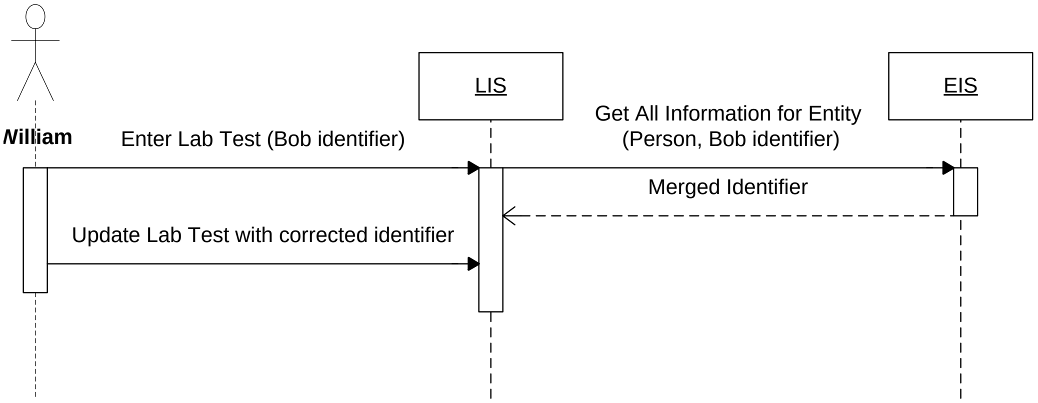
7.7.1 Single Entry Found


HIS
EIS
Greg
Search Bob
Find Entity With Traits(Person, Bob)
Return Bob
Locate Bob's Surgical Encounter
Enter Bob'sVital Signs

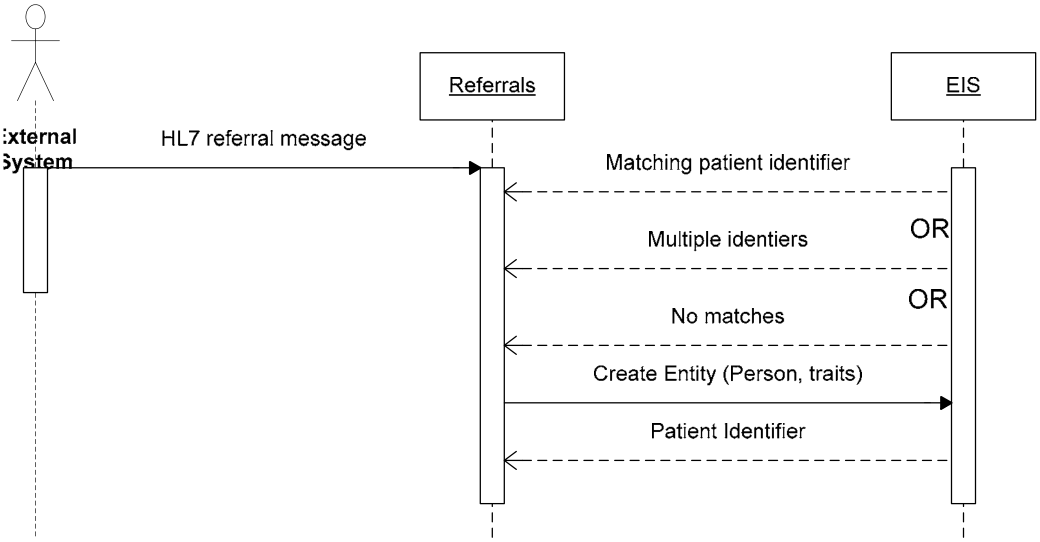
7.7.2 Multiple Entries Found


HIS
EIS
Greg
Search Bob
Find Entity With Traits(Person, Bob)
Return Person List Incl. Bob
Enter additional criteria
Find Entity With Traits
(Person, Bob, Age, Address)
Shorter Person List, incl Bob
Create new encounter

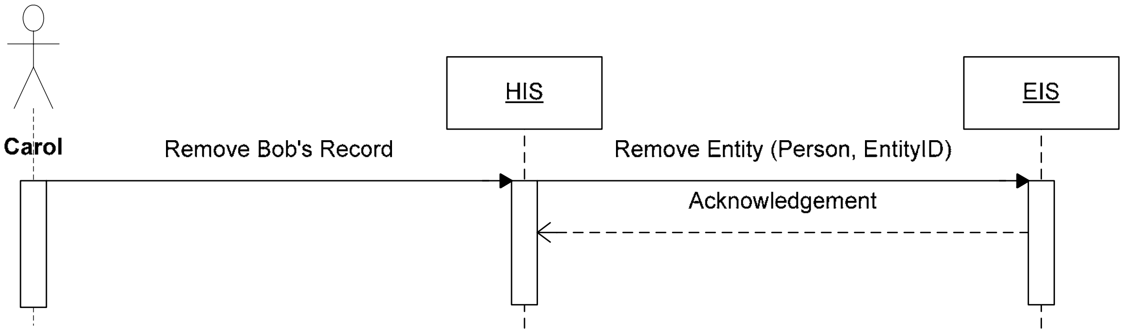
7.7.3 Merged Entries Found for an Identifier


LIS
EIS
William
Enter Lab Test (Bob identifier)
Get All Information for Entity (Person, Bob identifier)
Merged Identifier
Update Lab Test with corrected identifier

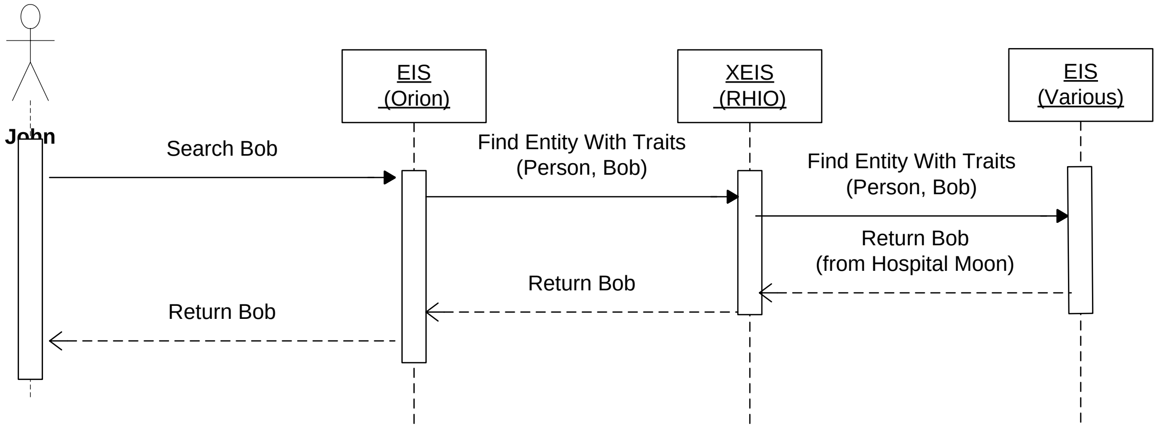
7.8 Unattended Encounter



7.9 Remove entity from the system

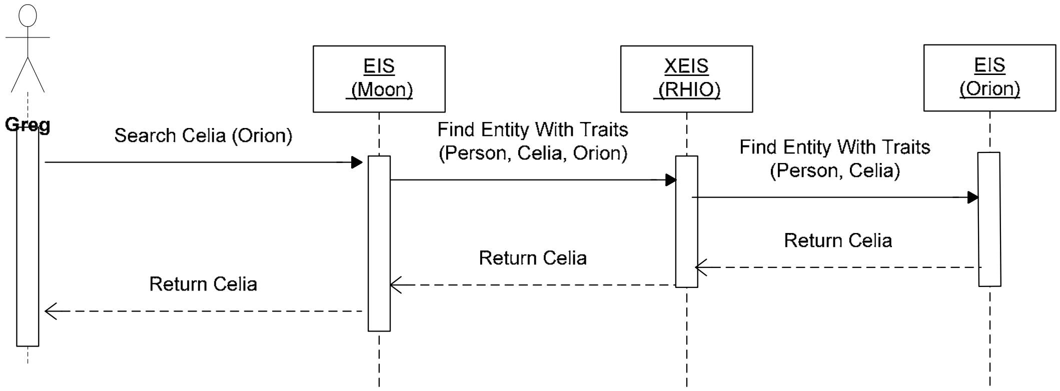


7.10 Look up patient across regional network

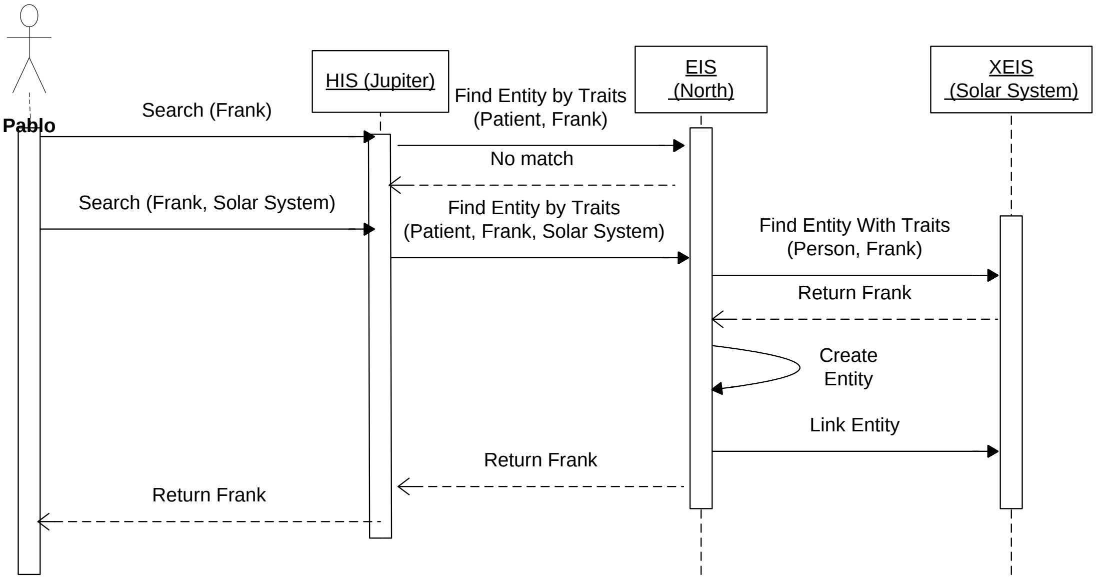


John
EIS (Orion)
XEIS (RHIO)
EIS (Various)
Search Bob
Find Entity With Traits (Person, Bob)
Find Entity With Traits (Person, Bob)
Return Bob (from Hospital Moon)
Return Bob
Return Bob

7.11 Look up a patient specifying a specific external organization



7.12 Link entities across regions within an organization



8 The Services Framework Functional Model

The Services Framework Functional Model identifies common underlying enterprise infrastructure such as naming, directory, security, etc. that may be assumed and referenced by this Functional Model.

Note that the Services Framework Functional Model is being developed in parallel with the service Functional Models; candidate functionality for the Framework should be submitted to the Infrastructure subgroup for evaluation.

9 Information Model and Semantic Binding Approach

The following principles shall be followed for specifying the information model to be used by the services being specified in this Service Functional Model:

1. SFMs shall provide a conformance profile supporting HL7 content where relevant
2. We shall not preclude the use of non-HL7 content
3. SFMs will reuse to the maximum extent possible the content models as defined in other standards (for example, HL7 RMIMs)
4. Information content representations shall be represented in platform-agnostic formalisms (e.g., UML)
5. SFMs may identify content at varying levels of granularity, depending upon the functions being specified. (For example, the Common Terminology Service will deal with different granularity of information than the Resource Location and Update Service).
6. Conformance Profiles may be balloted or adopted after the release of the initial SFM to address specialized business needs. (realm-specific profiles, domain-specific profiles, etc.)
7. Details about semantics specific to this SFM appear in other sections of this document

10 Recommendations for Technical RFP Issuance

The following issues needed to be considered when drafting the RFP and also by RFP Submitters. Where a specific section in this SFM mentions the issue, a reference has been included here so that further context may be obtained by the reader):

1. One specific semantic profile has been defined in this specification, i.e. the “HL7-V3-Patient” profile. The RFP or resulting submissions may choose to define additional profiles. It is suggested that a preference list is produced and included in the RFP when it is issued, which identifies which profiles should be included, or at least as “preference” criteria. This could include profiles for: Providers, HL7 V2 Patient, Non-living subjects etc. (Sections 2.3.1, 2.4, 6)
2. In the metadata section, individual “retrieve” operations have been defined. These could be combined with the “list” functions under the query interface. (Ref: Sections 5.2.3, 5.2.6, 5.2.9, 5.2.12)
3. Assignment, un-assignment and update of Traits to Entity Types. These have been defined as separate operations. They could just be treated as part of the Entity Type Update. (Ref: Sections 5.2.8, 5.2.13, 5.2.14, 5.2.15)
4. Trait Assignment to Entity Types has been defined as a single operation. The RFP submission could provide this as more of a bulk operation, but since this is likely to be infrequent, it has currently been left as individual updates, bearing in mind it allows for assigning to multiple domains in the one operation. (Ref: Section 5.2.13)
5. The use of a unique user-definable entity identifier has been treated as optional in this specification. This would be in addition to any identifying traits. It is left to the RFP Submitter whether this should remain, or whether to allow this to be dependent on the entity type. The operation descriptions assume it is optional, however several update and retrieval operations take the identifier as input. If this is not supported, then the RFP submitter must describe how this functionality will be achieved, e.g. using a system generated (hidden) identifier to achieve this purpose, which may require exposing the hidden identifier in some cases. In the case of the “unattended encounter” use case, generation of an identifier is really necessary. Identification of duplicates would also need to be considered. (Ref: Section 5.3)
6. Support for Entity Domains, Entity Types and Federation need to be discussed. See the discussion on federation in Appendix IV. This includes consideration of Cross Domain Entity Linking – RFP submissions should consider whether this operation should always be to and from a “master” domain for the XEIS instance, or allow peer to peer links. This is a behavior definition rather than explicit in the operation signature. (Ref: Sections 2.1.2, 2.3.2, 2.4, 5, Appendix IV)
7. Entity type specialization and query optimizations. The technical specification may include elements that are specific to an entity type, e.g. specific named operations for an entity type and also performance optimizations for specific queries. (Ref: Section 2.1.2, 2.4, 5.4.2)
8. Consideration needs to be given to further administrative functions in the technical specification. (Ref: Sections 2.3.1, 5.1)
9. Values, associated meaning and allowable actions on “Status” attributes for entities and also various metadata objects need to be defined (e.g. pending, active, deprecated, marked for deletion). (Ref: Section 5 – various operations)

10. In the Undo merge operation, rather than specifically requiring the Entity Identifier of the deprecated Entity, the implementation may offer an operation that shows recent merge transactions related to one Entity and reverse the transaction. (Ref: Section 5.3.5)
11. From an overall process perspective, the “unmerge” should require manual intervention afterwards. The RFP submitter may choose to enforce an inactive status on the unmerged entities until a further update is carried out. (Ref: Section 5.3.6)
12. The Link Entities interface permits linking of entities of different types, e.g. from a Person to a Patient. Submitters need to consider whether to restrict this to linking entities of the same type only. (Ref: Section 5.3.7)
13. The RFP submitter should discuss issues relating to service description and discovery.

11 Glossary

Citation of terms specific to this functional specification and not included in the overall HSSP Glossary

Term	Definition
Entity Domain	Within this specification, this identifies a sphere of control of entity identifiers. This could be legal (e.g. government issued identifiers), organizational (e.g. department, enterprise, cross-enterprise), geographical (e.g. regional, national, state) or even specific to one computer system. This describes the Entity Domain from a “usage” perspective rather than an “assigning authority” perspective as is used typically in HL7. See section 2.3.2 for a full discussion.
MPI / EMPI	(Enterprise) Master Patient Index. An application system that provides capabilities to find and cross-reference patients. MPIs will probably be the most common system to provide EIS interfaces.
Service Metadata	This is a set of data items that delineate the scope and coverage of EIS. This includes identifying the Entity Domains, Entity Types and Traits that an EIS supports.
Trait	This is a characteristic or attribute of an Entity that an EIS may use to identify Entities. An example would be “name” for a person.

12 Appendix I. Relevant Standards and Reference Content

This section identifies key existing standards that are relevant to this specification and how they are related to it. Given the diversity of work that has been carried out and continues to occur, not every standard or implementation can be referenced, so exclusion does not imply any qualitative comment or opinion. The purpose is to differentiate this specification from existing work and offer some guidance on how some of the key existing standards may play a part in implementations based on this specification.

Reference	Description	Relationship to EIS
HL7 Messaging Standard: V 2.5 and V3	The HL7 standard contains an excellent data model of relevant identifiers of persons.	HL7 models can and will be used to provide one or more of the semantic profiles to be used in EIS. It is not intended that any new domain analysis be carried out for specifying or implementing EIS. This is particularly relevant to models of "Persons" and their roles as patients and providers.
IHE Integration Profiles: PIX – Patient Identifier Cross-Referencing	Supports the cross-referencing of patient identifiers from multiple Patient Identifier Domains via the following interactions, including transmission of patient identity information from an identity source to the Patient Identifier Cross-reference Manager and the ability to access the list(s) of cross-referenced patient identifiers either via a query/ response or via update notification.	The IHE profiles basically coincide with conformance profiles of this specification, together with specific technology considerations. The relationship is described in detail in Appendix II. It is hoped that either through the OMG Submission process or independently, IHE will define additional profiles in the future to cover some of the additional EIS functionality and technology options.
IHE Integration Profiles: PDQ – Patient Demographics Query	Provides ways for multiple distributed applications to query a central patient information server for a list of patients, based on user-defined search criteria, and retrieve a patient's demographic (and, optionally, visit or visit-related) information directly into the application.	The IHE profiles basically coincide with conformance profiles of this specification, together with specific technology considerations. The relationship is described in detail in Appendix II. It is hoped that either through the OMG Submission process or independently, IHE will define additional profiles in the future to cover some of the additional EIS functionality and technology options.

Reference	Description	Relationship to EIS
IHE Integration Profiles: PWP – Personnel White Pages	Provides access to basic directory information on human workforce members. Supports two transactions, “find personnel white pages” and “query personnel white pages”.	The IHE profiles basically coincide with conformance profiles of this specification, together with specific technology considerations. In this case, the “find” transaction is out of scope for EIS, since it is effectively a service discovery mechanism. The “query” transaction is a constrained LDAP query, and LDAP is discussed below.
Network Working Group RFC3377, Sept 2002, Lightweight Directory Access Protocol (v3)	The Lightweight Directory Access Protocol (LDAP) is a fairly mature specification of a service for identifying people and entities and an interface for looking up people based on their properties.	The mechanism provided by LDAP is used primarily for authentication, but provides both domain (organization) scope and a fairly complete set of identifying traits for an individual. It is quite extensible, but doesn’t provide the querying functionality in terms of the quality of the match that might be required nor returning matches in any particular order. It does have full federation support and the ability to manage the traits. LDAP is used in many cases to handle entity identification. An LDAP look up could certainly provide part of the implementation behind an EIS interface, in a similar way as discussed for PIDS below.
Network Working Group RFC 2426, Sept 1998, vCard MIME Directory Profile	The vCard specification provides a set of identifiers for individuals and entities. vCard was used in the OMG PIDS specification as one of the sets of identifiers.	The vCard specification could be used for the traits in the EIS. There is no good mapping into the EIS domain. The vCard trait set could correspond to a particular semantic profile to be used in the EIS.
OMG Person Identification Service (PIDS) Specification Version 1.1, April 2001	A specification of a Person Identification Service with UML models and a complete specification in CORBA IDL. This includes most of the functions for managing a patient identity. The traits of an identity in this specification used the HL7 v2.3 and the vCard specifications.	This specification has been used as the basis for many implementations in different industries. This has been used as a primary source for many of the ideas in this specification. This specification could be seen as bringing PIDS up to date, although the additional “Entity” abstraction provides for additional flexibility of use.

Further details on relationship to PIDS:

This specification defines 'generic' interfaces that would allow name-value pairs to be associated with an entity, similar to its forerunner, the Person Identification Service (PIDS) module from the OMG. In fact, for person related functionality, the service interfaces defined in this document could be implemented over an implementation of the PIDS module as shown in the following diagram:

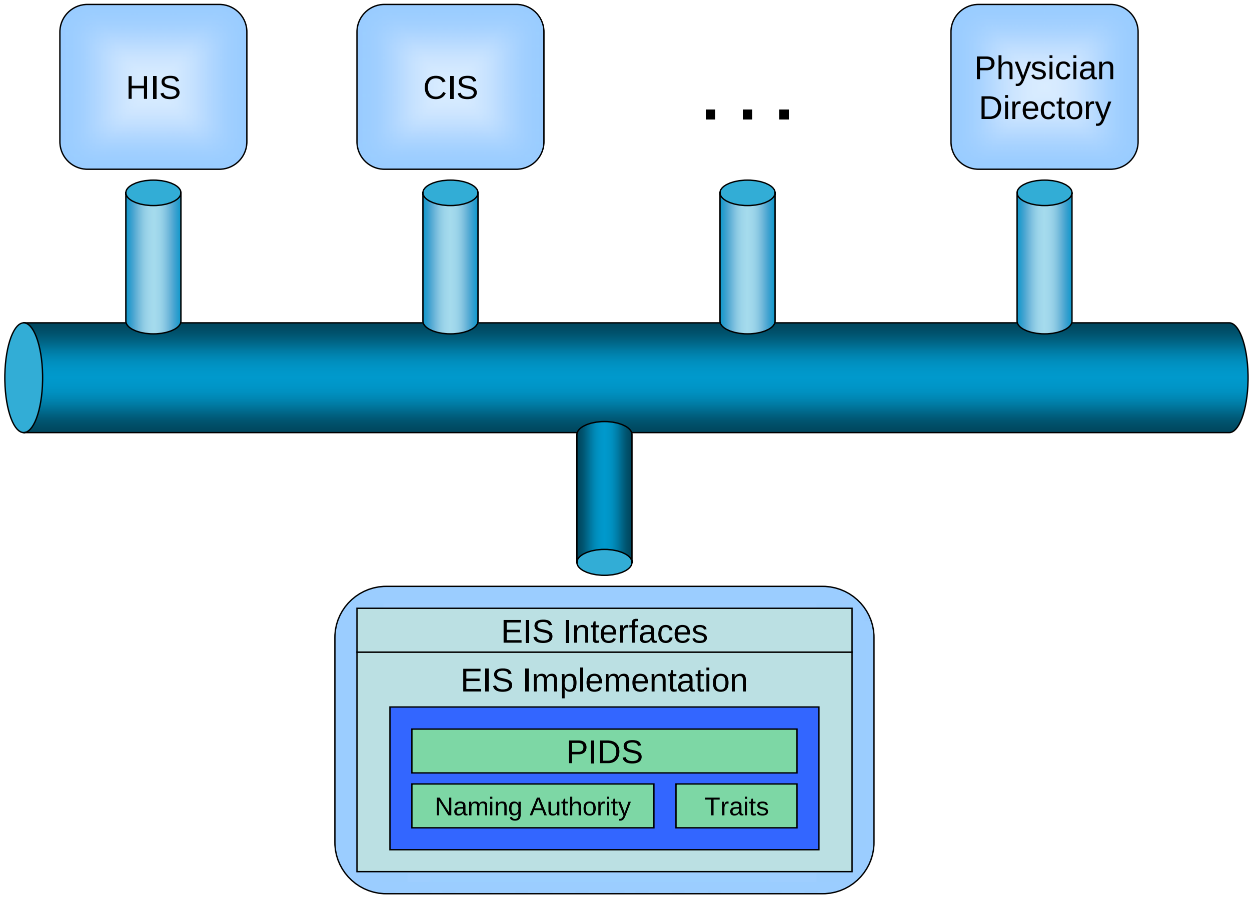


Figure 5 - EIS Relationship to PIDS

Finally, although not strictly related in the sens of overlap, it is worth mentioning the HL7 CCOW standard, which provide a mechanism for managing contexts across different applications within a workstation. A CCOW implementation could use an EIS implementation to manage and resolve entity identities, in which case would be a potential client of an EIS.

13 Appendix II – Relationship to IHE Profiles

(Note – as of time of writing, this is based on the HL7 V2 versions of the profiles. It is assumed that the V3 versions are functionally equivalent to the V2 versions, so this section would still apply. This section may be updated later if there are any functional differences)

Two IHE profiles are very relevant to the functionality provided by EIS, i.e. Patient Identifier Cross Reference (PIX) and Patient Demographics Query (PDQ). These profiles provide HL7 V2 and V3/web service based implementations of a subset of the functionality defined in EIS. With one exception (i.e. Patient Demographics Query with Visit information), these profiles can provide valid implementations of EIS “conformance profiles” (i.e. conformance profiles as defined in Section 6 of this document). Even some visit information could be added as traits, however the demographics query that returns visit information is really out of scope for an identity service, and more appropriately satisfied by one of the HSSP Retrieve, Locate, Update Service (RLUS) interfaces.

It is hoped that either through the OMG Submission process or independently, IHE will define additional profiles in the future to cover some of the additional EIS functionality and other technology options.

Overall, it should be noted that HSSP specifications such as EIS are “interface” specifications intended to describe service interfaces that would be implemented in a “services” layer within an overall architecture, and may support implementations of IHE profiles. Systems implementing IHE profiles may use EIS (or other) service layers to implement part or all of their functionality.

Each of the two profiles and the relationship to this specification is discussed in detail below. It should be noted that the section that discusses differences is NOT intended to be a qualitative evaluation. The sole intention is to point out differences, mainly which relate to scope, but does not imply any judgment of appropriateness. There will be many circumstances where the functionality provided by the scope of these profiles is wholly appropriate, and others where some of the other capabilities described in this Interface specification are appropriate. It should also be noted again that the SFM is an Interface specification, and therefore describes capabilities in those terms. Current implementations of systems that implement PIX/PDQ profiles may use other mechanisms (such as configuration files and/or code) to more statically implement some of the capabilities defined in the interfaces described in this SFM.

13.1 IHE PIX Profile (Patient Identifier Cross-Referencing)

The IHE PIX profile supports cross-referencing of patient identifiers from multiple Patient Identifier Domains. It has many live implementations, and provides a valuable means of ensuring interoperability for its defined scope. It explicitly supports three transactions:

- Transmission of patient identity information from an identity source to the Patient Identifier Cross-reference Manager

- The ability to access the list(s) of cross-referenced patient identifiers either via a query/ response
- Notification of updates of patient information

Since it is an “implementation” profile, it also defines specific technology solutions, initially HL7 V2 message based and more recently based on HL7 V3 messages encoded as XML and wrapped in SOAP envelopes. PIX implementations may also include pre-defined logic to implicitly cross-reference identifiers.

13.2 IHE PDQ Profile (Patient Demographics Query)

The IHE PDQ profile provides a method for multiple distributed applications to query a central patient information server for a list of patients, based on user-defined search criteria, and retrieve a patient’s demographic (and, optionally, visit or visit-related) information directly into the application. As for PIX, PDQ has many live implementations, and provides a valuable means of ensuring interoperability for its defined scope. It explicitly supports two transactions:

- Query of patient demographics
- Query of patient demographics and visit information

Since it is an “implementation” profile, it also defines specific technology solutions, initially HL7 V2 message based and more recently based on HL7 V3 messages encoded as XML and wrapped in SOAP envelopes.

From a functional perspective, the IHE PDQ profile bears similarity to a subset of the operations described in this specification and may be used to provide a profile for those operations (a sample of this is given below).

13.3 Differences between EIS and PIX/PDQ profiles

From a functional perspective, the IHE PIX and PDQ profiles bear similarity to a subset of the operations described in this specification and may be used to provide a profile for those operations (a sample of this is given below). However, there are a number of differences, as follows:

1. As stated above, PDQ supports an optional query that returns Visit information, which is beyond the scope of an identity service. This particular transaction could be provided by RLUS, and in theory could be provided using traits in EIS.
2. The PIX profile consumes HL7 messages not dedicated specifically to the purpose of maintaining the Identity service (e.g. patient update, admission) and derive the appropriate identity operation meaning from them. An additional adapter function would be needed to implement part of this service using a PIX profile, i.e. one that extracted the appropriate identity information and called the appropriate EIS operation.
3. The following capabilities are explicitly defined in EIS interfaces that are not supported by PIX or PDQ:

   a. EIS explicitly includes an interface to create and maintain entity cross references. PIX does not explicitly include this functionality.
   b. EIS provides an abstraction for supporting other entity types that uses the same operations and not just for patients.
   c. EIS includes specific operations for metadata management.
   d. EIS provides additional query operations.
   e. XEIS allows for the possibility of a peer-to-peer cross-referencing mechanism in addition to a central cross-referencing controller.

4. The EIS interfaces allow for any content model to be used, so e.g. OpenEHR Archetypes could be used to define the information content. This would be handled by the concept of “semantic profiles”.
5. Being a Functional Specification, EIS does not define the technology that will be used for implementation of interface or message content. Through the OMG RFP process (or otherwise), different technologies could be used to implement it, e.g. serialized Java objects. It also does not define some of the other aspects that are included in IHE, such as specific actors, which are part of the IHE Profile’s role as an “implementation” profile.

In terms of item number 3, it should be noted that not all implementations of EIS will necessarily support all of these capabilities, in fact all HSSP specifications support the notion of profiling as defined in Section 6 to enable implementations of subsets of the functionality.

Also, note that being a service interface description rather than an implementation description, this SFM does not explicitly define the functionality to carry out the “implicit” linking that is inherent in PIX implementations. However, it does describe the behavior in the notes section of the “create entity” and “update entity traits” operations. Since the exact behavior is dependent upon each individual implementation (i.e. subject to negotiation), it is not explicitly defined in this SFM.

13.4 Possible EIS Profiles for PIX and PDQ)

Bearing the above in mind, this (non-normative) section, suggests a set of profiles that could be used in order to provide a PIX/PDQ based implementation of part of this specification. Given that PIX/PDQ are technology specific, it is expected that such profiles (or similar) would be identified in OMG RFP submissions.

13.4.1 Functional Profiles:

- ➢ PIX Query
  - ➢ List Linked Entities
- ➢ PIX Entity Identifier Update
  - ➢ Create Entity

    - Update Entity Trait Values
    - Merge Entity
- PIX Notification Profile
    - Notify Entity Updates
- PDQ Entity Query
    - Find Entities by Trait

(Note – PDQ Patient Demographics and Visit Query returns visit information which is beyond the scope of this service)

13.4.2 Semantic Profiles

- HL7-V2.3.1-Patient: Patient as defined in HL7 V2.3.1
- HL7-V2.5-Patient: Patient as defined in HL7 V2.5

13.4.3 Conformance Profiles

Combining the functional and semantic profiles from above could give the following conformance profiles:

Functional Profile	Semantic Profile	**Conformance Profile**	Comments
PIX-Query PIX-Entity-Identifier-Update PIX-Notification	HL7-V2.3.1-Patient HL7-V2.5-Patient	**IHE-PIX**	Note that versions of PIX also imply a specific technology solution. It is reasonable to leave a single conformance profile at this level however and identify the variations at the RFP level.
PDQ-Entity-Query	HL7-V2.5-Patient	**IHE-PDQ**	Note that versions of PDQ also imply a specific technology solution. It is reasonable to leave a single conformance profile at this level however and identify the variations at the RFP level. This service would not support patient visit information.

In addition to the above, the conformance profile would also need to indicate that the automated matching is required – for create and update entity operations and also considered for merge.

14 Appendix III - HL7 EHR Functional Model Traceability

This section lists the EHR Functions that are related to this service.

Note that in general there will not be a direct correspondence between EHR Functions and HSSP Services, since Services are specified from a different system viewpoint. The mapping provided here enables the HSSP Services to be understood in the context of the ***EHR-S Functional Model DSTU****.*

EHR Function ID	EHR Function Name	EHR Function Statement	Notes *For every row, explain the rationale for including in this specification.*
DC.1.1.1	Identify and maintain a patient record	Identify and maintain a single patient record for each patient.	This EHR functionality is precisely what this service addresses.
DC.1.1.2	Manage patient demographics	Capture and maintain demographic information. Where appropriate, the data should be clinically relevant, reportable and trackable over time.	In order to maintain a single patient record, the demographics for the patient must be stored in order to be used to find the patient. While clinical data is not specifically supported in the EIS, clinical traits for an entity could be defined and managed.
S.1.3*	Provider directory	Provide a current directory of practitioner, team, department, organization, etc., information in accordance with relevant laws, regulations, and conventions.	The EIS could support this functionality through the use of the 'find' interface. Assumes that a trait for each entity will uniquely define the role the entity assumes – patient or provider.
S.1.4	Patient directory	Provide a current directory of patient information in accordance with relevant privacy and other	The EIS could support this functionality through the use of the 'find' interface. Assumes that a trait for each entity will

		applicable laws, regulations, and conventions.	uniquely define the role the entity assumes – patient or provider.
S.1.4.1	Patient demographics	Support interactions with other systems, applications, and modules to enable the maintenance of updated demographic information in accordance with realm-specific recordkeeping requirements.	The EIS is intended to resolve issues derived from having multiple systems, modules and applications storing and accessing different sources of demographic information.
S.1.4.3	Patient's residence for the provision and administration of services	Provide the patient's residence information solely for purposes related to the provision and administration of services to the patient, patient transport, and as required for public health reporting.	The demographics information stored in the EIS will cover this requirement.
S.3.5*	Subject to Subject relationship	Capture relationships between patients and others to facilitate appropriate access to their health record on this basis (e.g. parent of a child) if appropriate.	Through the use of traits for the entity, the relationships with other entities could be established, although this would need further consideration.
I.1.1	Entity Authentication	Manage the sets of access-control permissions granted to entities using the EHR-S. An EHR-S controls authorizations to entities that use the EHR-S, for roles, and within contexts. A combination of the authorization levels may be applied to control access to EHR-S functions or data.	The EIS will provide the means to identify a particular patient and then could associate access level controls to that patient's information.
I.2.1	Data Retention, Availability and Destruction	Retain, ensure availability, and destroy health record information according to organizational standards. This includes: Retaining all EHR data and clinical documents for the time period designated by policy or legal requirement; Retaining inbound documents as originally received (unaltered); Ensuring availability of information for the legally prescribed period of	While the EIS, by itself, is unlikely to provide the complete solution for this function, the EIS may contain traits for birth, death and an audit log for changes to the record to reflect those dates. In addition, the EIS will be the entry point to the RLUS to access the clinical records an associated audit logs.

		time; Providing the ability to destroy EHR data/records in a systematic way according to policy and after the legally prescribed retention period.	
I.2.2	Audit trail	Provide audit trail capabilities for resource access and usage indicating the author, the modification (where pertinent), and the date/time at which a record was created, modified, viewed, extracted, or deleted. Audit trails extend to information exchange and to audit of consent status management to (to support DC.1.5.1) and to entity authentication attempts. Audit functionality includes the ability to generate audit reports and to interactively view change history for individual health records or for the EHR-S.	The EIS must provide change logging to an audit manager to support this function.
I.2.3	Synchronization	Maintain synchronization involving: Interaction with entity directories; Linkage of received data with existing entity records; Location of each health record component; Communication of changes between key systems.	The EIS must provide the ability for change negotiation, especially in cases of XEIS deployments. The ability to provide identity resolution for a person when that person's demographics have changed are dependent on correct distribution of changes.
I.3	Unique identity, registry, and directory services	Enable secure use of registry services and directories to uniquely identify and supply links for retrieval and identify the location of subjects of care and providers for health care purposes; payers, health plans, sponsors, employers and public health agencies for administrative and financial purposes; health care resources and devices for resource management purposes.	This is the core functionality provided by this service.

15 Appendix IV – Federation and use of Entity Domains and Entity Types

Note - This Appendix should be treated as “informative”, i.e. input for consideration by RFP submitters.

As has been alluded to a number of times within this specification, the capabilities have intentionally been defined at an abstract level (i.e. use of the generic “Entity”). This ensures that RFP Submitters and resulting implementations are given the freedom to provide as much implementation flexibility (or as little) as they need while still maximizing interoperability. It is important that this flexibility is supported functionally, and then to constrain this at the technology specific level. It is considerably more difficult to introduce it further down the stack. This also ensures the maximum level of consistency across different technical and information specific implementations.

As stated earlier, the interfaces and operations above are defined on the basis of an XEIS that manages multiple Entity Domains and multiple Entity Types. A number of different “levels of support” for multiple Entity Domains and Entity Types are described below, together with tables depicting how the operations would be different at each level.

15.1 Federation Concerns

One issue with respect to Entity Domains is how they are identified and managed, particularly in cross-organization scenarios. In order to coordinate identities across any set of Entity Domains, there is usually a higher level Entity Domain that sets the rules, forming a hierarchic pattern, e.g. National -> Regional -> Department etc. However, this specification acknowledges the fact that peer-to-peer scenarios may become increasingly common, and even those where the contract negotiations are dynamic. This would allow several organizations to interact and share identifying information without assuming the presence of an overall control Entity Domain. Both the hierarchic and peer-to-peer topology are supported by the interfaces in this specification. Even in the peer-to-peer case, there still needs to be a mechanism to agree on the actual Entity Domain identifiers, which could be some controlling authority or by contractual agreement.

Issues relating to scope of individual EIS and XEIS instances are discussed further below, but whatever the scope, any XEIS and most EIS instances will be able to take part in federations. In the case of an EIS, this can be viewed as a “child” in a hierarchic relationship, usually with a single controlling XEIS. In the case of an XEIS, it could take part in any kind of relationship, i.e. peer-to-peer, hierarchic (upwards and/or downwards) or combinations of both. (Note - The implementation mechanisms for achieving this are beyond the scope of this functional model, although current work in the overall HSSP project is investigating a namespace-based approach within the Service Development Framework for input to RFPs.)

15.2 Use of Entity Domains

Again as stated earlier, allowing a service instance to support multiple Entity Domains provides a de-coupling between organizational structure and software topology and distribution, one of many aspects of de-coupling that is important within a Service-Oriented Architecture. For the purposes of this discussion, in terms of support for Entity Domains, four main levels have been identified:

1. Full Multiple Entity Domain Support (XEIS-Full)
2. Multiple Entity Domain Support with a fixed “owning” Entity Domain (XEIS-Single)
3. Multiple Entity Domain Support -Entity linking and unlinking only (XEIS-Link)
4. Single Entity Domain Support (EIS)

These may be defined as functional or technical profiles within the RFP Submissions. Each of these is considered further below. Note that there may well be other levels or combinations that may be implemented, but by covering these four, it should be easy to derive what the implications would be for other combinations:

Full Multiple Entity Domain Support (XEIS-Full)

This form of XEIS allows a single instance to manage entity identifiers in multiple Entity Domains. Entities from different Entity Domains could be created and managed, including setting up cross Entity Domain links between them. This is facilitated by the presence of the Entity Domain related metadata management operations and the presence of the Entity Domain Identifier in many of the entity level operations.

An alternative implementation, which avoids using the Entity Domain Identifier as a specific parameter in entity level operations would be to use the Services Registry look up to differentiate between physical “endpoints” that support different domains, which would all use the same interface definition. On the one hand this would create administrative / configuration management difficulties but it does avoid the use of the domain parameter, which could allow for more consistency with the other three options. With the increasing flexibility and dynamic behavior encouraged by Service Oriented Architectures, the explicit use of the Identifier has been defined in the operation definitions to enable simpler management. In environments where the Entity Domains themselves may change frequently, then this would be of some advantage, in more static scenarios less so. RFP submissions must take a stance on which methods they support. Also, a similar approach could be taken for dealing with different Entity Types within a single EIS instance. Supporting both multiple Entity Domains and multiple Entity Types using the separate endpoint approach could become very difficult from a systems management perspective.

Another alternative could be for XEIS/EIS instances to treat these items as just additional “traits”, although this is not recommended since these may directly affect the overall behavior.

In cross domain scenarios, in the “Find Entities by Trait” operation the use of explicit Entity Domain parameters is necessary. This occurs when a Service Consumer (XEIS, EIS, HIS etc) wishes to request information on linked entities from a specific set of Entity Domains managed by another XEIS. See the diagram below. So, for example, if an HIS is only able to connect to D (its local EIS instance) and wishes to ask for Entity information managed by B and C only but not A, this would not be feasible without an explicit Entity Domain parameter. As sharing of medical information across organization boundaries becomes more common, this is likely to become a very necessary capability. It is very likely that an organization may wish to enquire of a network (such as a RHIO) for patient identities from certain specific other organizations or geographic locations, particularly for time sensitive queries or where the patient provides specific information. It may even be a consent issue if a patient is willing for an organization to be able to obtain his/her information from certain specific other sources but not others.

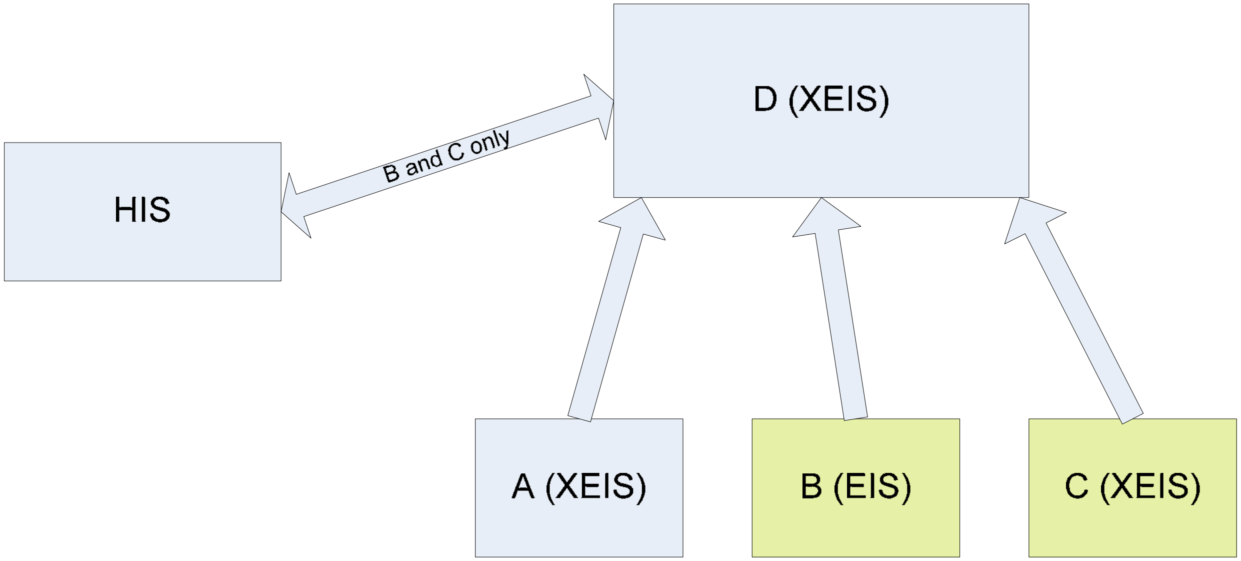


Figure 6 - Full Multiple Entity Domain Support (XEIS-Full)

Multiple Entity Domain Support with fixed owning Entity Domain (XEIS-Single)

This assumes a hierarchic structure of XEIS, where any one instance has a single managing Entity Domain and provides cross referencing to other child Entity Domains. Entities and traits may be added and maintained in the managing Entity Domain, as well as providing cross references to other child Entity Domains. In this case, a specific Entity Domain Identifier is only relevant in the Link and Unlink operations. The Entity Domain for all other operations would be fixed to the owning Entity Domain. In this case, there are three main implementation level options for restricting the Entity Domain:

1. Keeping the Entity Domain in the operation parameters but fixing the value as constant.

   This has the advantage of allowing for migrating to multiple domain support at a later stage with minimal change. At the same time, this includes an unnecessary parameter in many operations.

2. Removing the Entity Domain from the operation parameters.

This has the advantage of simplicity, but makes migration more difficult and would result in a different interface than for the Full XEIS.

3. Defining the Entity Domain parameter as optional

   This may be a reasonable compromise, and has the advantage of a consistent interface definition. It does however lead to the need to add in additional validation logic to deal with the conditional behavior.

Multiple Entity Domain Support – Entity linking and unlinking (XEIS-Link)

This provides a limited set of functionality (more like the IHE PIX Profile). This basically implements the Entity Cross-Reference functional profile identified in Section 6 together with a subset of the Entity Update functions. This may be useful in situations such as networks where the network is providing a way of connecting multiple organizations without managing any Entities in its own right. Note that this could be implemented in such a way to maintain peer-to-peer cross references or structure them in a hierarchy so effectively implementing a restriction of either XEIS-Full or XEIS-Single.

Single Entity Domain Support (EIS)

In this case, only one Entity Domain is known to the EIS. The issues discussed in XEIS-Single above apply here to, but an EIS cannot support any cross- Entity-Domain linking since it has no real notion of Entity Domain. If an EIS delegates a query to an XEIS, it may still pass on the requested Entity Domain identifiers to the XEIS.

Analysis of Relevant Functional Operations for each type of EIS:

The table below cross-references some of the operations from this specification against the four different levels of EIS identified above. Only operations relevant to this discussion are listed. The others are not affected by this discussion. (Where the Entity Domain Identifier is stated as not required, the implementation options as described above under XEIS-Single would apply). In this table, the XEIS-Link is assumed to be based on XEIS-Single in the table below (i.e. hierarchy based). Variations in output are not considered.

Operation	XEIS-Full	XEIS-Single	XEIS-Link	EIS
Metadata Interface:				
Create/Update/Retrieve a Entity Domain	Support Required	Optional (would allow for setting restrictions for access to particular Entity Domains)	Optional (would allow for setting restrictions for access to particular Entity Domains)	Not supported
Create/Update/Retrieve an Entity Type	If multiple entity types supported:	If multiple entity types supported:	If multiple entity types supported:	If multiple entity types supported:

Operation	XEIS-Full	XEIS-Single	XEIS-Link	EIS
	Entity Domain input required	Entity Domain input not relevant	Entity Domain input optional	Entity Domain input not relevant
Assign / Un-assign / Update Trait Assignment	Entity Domain input required	Entity Domain input not relevant	Entity Domain input optional	Entity Domain input not relevant
Entity Management Interface:				
Create / Update / Activate / Inactivate / Merge / Unmerge / Undo Merge Entities / Update Trait Values	Entity Domain input required	Entity Domain input not relevant	Entity Domain input optional	Entity Domain input not relevant
Link / Unlink Entities	Two Entity Domain inputs required	Only Entity Domain-2 input required	Entity Domain-1 optional, Entity Domain-2 required	Entity Domain input not relevant
Query Interface:				
Get All Information for Entity / Request update notifications /	Entity Domain input required	Entity Domain input not relevant	Entity Domain input optional	Entity Domain input not relevant
Find Entities by Trait / List entities by entity type and domain	Entity Domain input optional	Entity Domain input not relevant	Entity Domain input optional	Entity Domain input not relevant
List Linked Entities	Main Entity Domain required. Qualifying Entity Domains optional	Main Entity Domain not relevant. Qualifying Entity Domains optional	Main Entity Domain optional. Qualifying Entity Domains optional	Entity Domain not relevant.
List Supported Entity	Support Required	Optional	Optional	Not supported

Operation	XEIS-Full	XEIS-Single	XEIS-Link	EIS
Domains				

15.3 Use of Entity Types

A similar set of concerns also applies to the use of Entity Types. For those EIS instances that deal with only a single Entity Type, the presence of Entity Type parameters would add unnecessary overhead. At the same time, concerns of consistency, flexibility and future migration suggest that it should be included in all cases.

As with the Entity Domain, there appear to be the same two main options where multiple Entity Types are supported (parameter based or use of separate interfaces with registry look-up) and the same three main options for those cases where multiple types are not supported (constants, no parameters or optional). Rather than repeat the discussion again, the reader can simply substitute Entity Type for Entity Domain in the discussion above. As with EIS and XEIS, some instances will explicitly support only one underlying type, and this will not change. However, even in those cases, the use of multiple entity types could provide an effective solution for support of different semantic profiles, e.g.

- Version management of information models, so e.g. HL7 V3.0 definition of Patient and V3.1 would be defined as different entity types and could be supported concurrently without an interface version change.
- Concurrent support for different information models, e.g. HL7 V3 and Open EHR Archetypes

Almost certainly, the most common example will be the use of EIS for Patient information, and this has provided the basis for the scenarios and examples in this document. However, other instances may support identification of many different resource types in a much more dynamic way. The interfaces as defined support a full range of this variation, but specific semantic profiles should be defined and the technical specifications must be very clear as to how these concepts will be physically supported.

16 Appendix V – Implementation of a “Simple” EIS

Note - This Appendix is also “informative”, i.e. input for consideration by RFP submitters and/or implementers.

The capabilities described within this functional specification are deliberately flexible and comprehensive. In particular, as mentioned earlier, support is defined for multiple Entity Domains and multiple Entity Types as well as an explicit metadata maintenance interface. However, it is feasible that many implementations would be far simpler, e.g. single entity type (Patient) and single Entity Domain and fixed or static metadata. There would be various ways of achieving this, some which conform to the letter of this specification, and others which conform to the spirit.

This section briefly outlines several possibilities for achieving such simpler implementations, which could be applied on their own or in combination (there may well be others):

- Use of Profiles for constraining function and entity types
- Fixed Entity Type and Specific Operation Naming
- Configuration Files for static metadata

16.1 Use of Profiles

This is similar to the discussion on the IHE Profiles in Appendix II. Note that “standard” profiles can be defined through the HSSP process (at the functional or implementation level) or independently by single organizations or groups of organizations for their own interoperability domains. Based on a simple scenario where HL7 V3 Patient information is supported with only a subset of the operations, the following (sample) profiles would be used:

Functional Profile:

- Simple Entity Identifier Update
  - Create Entity
  - Update Entity Trait Values
  - Merge Entity
- Simple Entity Query
  - Find Entities by Trait

Semantic Profile:

- HL7-V3-RIM-V2-14-Patient: as defined in section 6 above

Conformance Profile:

- Simple Patient Profile = Simple Entity Identifier Update + Simple Entity Query + HL7-V3-Patient

This option provides a fairly simple mechanism for providing a well-defined subset of the overall capabilities.

16.2 Fixed Entity Type and Specific Operation Naming

This simply means locking the Entity Type and/or actually renaming the operations with the explicit entity type name rather than the generic entity. This certainly remains within the spirit of the functional specification, and may be considered by implementers or RFP submitters. So, for example, instead of “Find Entities by Trait”, the operation would be “Find Patient by Trait” and so on. This could be defined as, or combined with a specific profile definition. Note that the use of the “Entity Type” as an explicit parameter was discussed in Appendix IV.

16.3 Configuration Files for Metadata

A set of operations were defined in Section 5 to provide capabilities to manipulate the metadata used by the service. This provides a very dynamic capability whereby service instances can effectively be re-configured to deal with different Domains and/or Entity Types and/or Trait sets. As usual however, such flexibility comes with a price tag, so a fairly easy alternative is either to fix the values where such flexibility is not required, or to use more static “configuration files” to define the supported metadata. Either alternatives are valid implementation options.