Integrating the Healthcare Enterprise

IHE IT Infrastructure
Technical Framework Supplement

Care Services Discovery
(CSD)

Trial Implementation

20 Date: September 9, 2016

Author: IT Infrastructure Technical Committee

Email: iti@ihe.net

Foreword

This is a supplement to the IHE IT Infrastructure Technical Framework V13.0. Each supplement
30   undergoes a process of public comment and trial implementation before being incorporated into
the volumes of the Technical Frameworks.

This supplement is published on September 9, 2016 for trial implementation and may be
available for testing at subsequent IHE Connectathons. The supplement may be amended based
on the results of testing. Following successful testing it will be incorporated into the IT
35   Infrastructure Technical Framework. Comments are invited and may be submitted at
http://www.ihe.net/ITI_Public_Comments.

This supplement describes changes to the existing technical framework documents.

"Boxed" instructions like the sample below indicate to the Volume Editor how to integrate the
relevant section(s) into the relevant Technical Framework volume.

40   *Amend Section X.X by the following:*

Where the amendment adds text, make the added text **bold underline**. Where the amendment
removes text, make the removed text **bold strikethrough**. When entire new sections are added,
introduce with editor's instructions to "add new text" or similar, which for readability are not
bolded or underlined.

45

General information about IHE can be found at: http://www.ihe.net.

Information about the IHE IT Infrastructure domain can be found at:
http://www.ihe.net/IHE_Domains.

Information about the structure of IHE Technical Frameworks and Supplements can be found at:
50   http://www.ihe.net/IHE_Process and http://www.ihe.net/Profiles.

The current version of the IHE Technical Framework can be found at:
http://www.ihe.net/Technical_Frameworks.

CONTENTS

55

180

Introduction to this Supplement

This introduces the Care Services Discovery (CSD) Profile. The CSD Profile supports queries across related directories containing data about: organizations, facilities, services and providers. Queries against an optional "FreeBusy" service are also supported; this FreeBusy information would support the development of a list of schedulable time slots for providers or services at specific facilities.

The CSD Profile describes four actors and the transactions between them:

1. **Service Finder** – the Service Finder Actor submits queries to the Care Services InfoManager, who returns the requested result(s). Queries may be expressed as invocations of "stored" queries, or (optionally) as ad hoc queries using the flexible XQuery standard, or (optionally) as iCalendar vFREEBUSY queries.

2. **Care Services InfoManager** – the InfoManager Actor has two roles. The InfoManager maintains a local cache of content that represents cross-referenced data from one or more Care Services Directory actors. The InfoManager periodically refreshes this cache by querying Directory actors for updated content. The InfoManager's other role is to process inbound queries from Service Finder actors. These queries are executed against the cached content.

3. **Care Services Directory** – the Directory Actor is responsible for returning an XML document in response to a refresh request from a Care Services InfoManager. The response document contains content which has been inserted or updated in the directory since the last refresh and is expressed in the format defined by the CSD Profile's XML schema.

4. **Service Availability** – the Service Availability Actor responds to requests for the "busy" time for a provider offering a service at a specified facility or for a service at a specified facility. This busy data may be employed to determine the free time slots which may be scheduled. The format of the request and response is defined by the IETF CalDAV specification (RFC 4791).

Because it maintains interlinked directory information, the CSD Profile is able to respond to queries such as:

- Which facilities are associated with which organizations?

- What services are provided at specific facilities or, conversely, where are the facilities that provide a specified service?

- Who are the providers associated with a particular organization; what services do they provide; at which facilities do they provide these services, and when?

- Within a specified date range, when are the schedulable time slots for the provider of a specific service?

The CSD Profile's loosely coupled design and flexible querying capability means it can be deployed within a number of eHealth architectures and support a wide array of care workflows.

220   It is contemplated that a future iteration of this profile will define how the Care Services InfoManager Actor will support an interoperable library of store queries which may be specified by an implementing jurisdiction. These additional stored queries would be expressed using the same XML schema and XQuery expression format as has been employed to describe the four mandatory stored queries described in sections ITI TF-2c: 3.73.4.1.2.1 – 3.73.4.1.2.4.

225   It is also contemplated that a future iteration of the profile will support ATOM feeds in addition to the current mechanism for result set paging that is supported in this version of the profile. Such support will entail the addition of a result set caching capability on the InfoManager plus a new GET message which may be employed by the Service Finder to navigate through this cached result.

Open Issues and Questions

- CSD013:

  - Should the implementers of a CSD service be required to develop/deploy a monolithic directory structure? Even if an LDAP query format is defined by this profile (as is the case with HPD, for instance), it does not require that the underlying data structure is an LDAP directory (there are LDAP to SQL translators, for example). What if a federation of independent registries are together filling the role of a Care Services Directory Actor? To support federation – the Care Services InfoManager will need to know what the address of each of the federated directories is.

  - Addressed (13-05-14): Adopting XQuery in favor LDAP addresses this – but it remains an open issue to see if the XSD is, in fact, a superset of the LDAP structure. **Did we "break" the HPD information model? Is CP 601 appropriately supported?**

- CSD019

  - Should there be a mandatory relationship between provider and facility or provider and facilityService? The issue of modeling "community health worker" (CHW) or "home nurse" is particularly germane. If such a model is to be constructing using "facility" as a proxy for catchment area, then there are advantages to the mandatory relation. Otherwise, the mandatory relationship to organization may be preferred. Essentially, **how will CSD support a query that displays: how many CHWs are active within a specified catchment area?**

- CSD020

  - What should be the appropriate Status values for organization, facility and provider? Should this list be mandatory (CWE) but with allowed extensions? **Should the coding be left entirely up to implementers?** Backward compatibility with HPD is an issue, since status was specified in that profile. Different status values may be specified by an implementer for each of these; also – another status attribute may be developed and included in the XSD.

- CSD031

  - As an informative appendix, a mapping between ITI-58 and XQuery is recommended. Change (13-06-07): it is no longer expected that a Consumer will be able to submit ITI-
260 58 transactions to a Care Services InfoManager. Instead, it is expected that it will be simpler to support an HPD Directory Actor as a Care Services Directory Actor. **It remains an open issue to develop a way to map HPD-conformant LDAP content to a CSD-conformant XML document so that it can be returned in response to a Care Services InfoManager refresh poll.**

265 ## **Closed Issues**

- CSD001:

  - Issue: Investigation of available specifications in this area identified the OMG ServD standard (in development). This specification usefully addresses a number of the use cases important to CSD but did not leverage the existing IHE HPD specification which is
270 based on LDAP and its extensions (PWP, ISO/TS 21091).

  - Resolution: (13-03-18) Based on the research that was done, it was decided that HPD could be expended to support the CSD use cases and that this approach was to be preferred.

- CSD002 (HPD004):

275 - Issue: Do we need to include "Provider Privileges at an organization" as an attribute to be defined in this standard?

  - Resolution (13-03-18) – "Provider Privileges", where these map to services, may be indicated by establishing a member-of relationship between an Individual Provider and a FacilityService. This is not directly analogous to admitting privileges and is not intended
280 to be. Rather, it is intended to indicate where a clinician provides a specific service at a specific location. In this way, a Service Finder could query for all the FacilityServices for a specific provider.

- CSD003 (HPD007):

  - Issue: How will we identify the limited number of locations of a provider, if the provider
285 works at a subset of the locations of an organization?

  - Resolution (13-03-18): As indicated above, an individual provider may have a member-of relationship with multiple FacilityService entities. The provider will inherit the address from these which can optionally be extended /overridden to indicate the "office hours" indicative of when the provider is available to provide the indicated services there.

290 - CSD004 (HPD008):

  - Issue: How will global identifiers be handled? This discussion came up as a result of a discussion on NPI numbers.

- Resolution (3/15/10) – Each Organization, Provider, Facility, and FacilityService will have at least ONE globally unique identifier. It is anticipated that the mandatory unique identifier will be system-generated (e.g., a GUID).

- CSD005 (HPD009):

  - Issue: Should the Validation attribute structure be included in the schema for the initial profile, or be extended in a later update of the profile? This attribute structure indicates whether or not validation of this information has been done (Flag), when the validation was done (Date), who did the validation (Source). The validation attribute structure can be included for multiple areas in the schema, but most importantly for overall validation of the Organization and Individual Provider attributes, the Relationship information, and the Certification, Specialty, and Degree information.

  - Resolution (4/26/10): The validation attribute will NOT be included in the schema for the initial profile. It is assumed that the HPD Actor will validate the feed. That needs to be clearly stated in the document (see Section 35.1.1.3). NOTE: This mechanism to address the issue is directly referenced in the Use Cases.

- CSD006 (HPD0010):

  - Issue: Currently, language, as a multiple attribute, is associated with Individual Provider only. Should language be considered an attribute for Organizational Provider? A Use Case has not been defined for this.

  - Resolution (4/26/10): Language should be considered an attribute for Organizational Provider as well as Individual Provider. A Use Case has been added to reflect this. NOTE: this is an important CSD Use Case in a developing country context (languages spoken at a Facility).

- CSD007 (HPD0011):

  - Issue: How should addresses be defined so that the definition meets global needs?

  - Resolution: (13-03-18): To support the CSD Use Cases, it will be necessary (MUST) to support geocodes in the Address object.

- CSD009 (HPD0016):

  - Issue: Currently *Specialty Role* has been included as an attribute for Organizational Provider. Should this be kept?

  - HPD0016 Resolution (4/07/10) - There is no Use Case for this and the Specialty Role changes too frequently. Do not include it.

- CSD009 Resolution (13-03-18): Each Organization may have Facilities; these Facilities may have FacilityService entries. The relationship between Organization and FacilityService indicates what services are being offered by an Organization, albeit through two levels of a hierarchy.

- CSD008 (HPD0013):

330   • HPD0013 Issue: The following Category attributes do not have agreed to value definitions:

- Identifiers

- Addresses

- HPD0013Resolution (4/22/10):

335   • Identifiers – The values will be defined by national or regional organizations.

- Addresses – This profile only addresses three address types, each of which is a separately defined attribute in the auxiliary class (Billing Address, Mailing Address, and Practice Address).

- CSD008 Issue & Potential Resolution (13-03-18):

340   • The manageability of addresses is a concern. It is considered that, for CSD, a Facility MUST have a practice address. This Facility address MUST be inherited by any related FacilityService and may be constrained by the FacilityService (perhaps in the member-of entry?) only regarding business hours. Resolution (13-04-30): an association class is being used to model the relationship between Facility and Service.

345   • As above, where a provider has a relationship to a FacilityService, the practice address MUST be inherited and, likewise, may be constrained regarding business hours. Resolution (13-04-30): an association class models the relationship between Providers and FacilityServices.

- Related to this… should an Organization be prevented from having a Practice
350   Address (or should it be ignored if it is there)? It is considered that this would be a useful constraint that would improve manageability. Resolution (13-04-30): only administrative addresses are modeled for Organizations.

- How will mobile clinics be handled? Are these facilities (likely YES). Is the address sufficiently flexible that a facility can have address as a required field but the
355   construct can still be used to describe a mobile facility? Resolution (13-04-30): omit mobile facilities from this profile in this iteration. There are ways to add mobile clinics (later) be making address optional.

• CSD010 (HPD0018):

- Issue raised during a discussion of Section 3.73.4.1 Provider Feed Request: As of 4/20/10
360   the profile reflects the decision made at the F2F in February that for a Provider Feed request, the only response provided by the HPD Actor would be an acknowledgement that the transaction has been received. Data administration issues such as data reconciliation, data validation, data integrity etc. associated with the Add/Update/Delete operations are considered back-end processes for the purpose of this profile and proposed
365   to be addressed by the policies and procedures of the organization managing the HPD. The profile would not provide any guidance on any expected actions or back-end processes to be executed, or policies to be followed by the HPD Actor. If that is the case

then this transaction would be an optional transaction. This may have been misunderstood, or an invalid conclusion and needs re-addressing by the IHE committee.

370
- HPD0018 Resolution (4/26/10): This is a required transaction for the HPD Actor. The required actions for the HPD Actor are that the HPD Actor is required to have a policy to conform to LDAP specification for processing adds, deletes, and updates. The HPD Actor must have a policy to validate feeds and handle data integrity prior to publishing the date.

375
- CSD010 Issue: Will CSD support, via the Provider Feed (for example), management of Facility and FacilityService entries? Resolution (13-04-30): data management will not be supported in this iteration of the CSD Profile.

- CSD011 (HPD024):

  - HPD024 Issue: Use of Address Structure:

380
    - First Option is to make an address as an attribute with syntax dstring *("$" dstring) similar to that of the postal code but additionally enforce format of "key=value" that allows for key to be of different kinds: address status and address components. This would allow adding new types without redefining the schema and allow us to search for &(status=primary, city=Nowheresville).

385
    - Second option is to have two different attributes as LDAP Postal Address syntax to distinguish primary and other addresses. E.g., *hpdPrimaryProviderPracticeAddress* and *hpdProviderPracticeAddress.* In this option, the Address Status attribute is not maintained and assumes Primary Address as always active.

390
    - Third option is to search based on a value stored in a subordinate Address object. In that case, we have to obtain that object and then perform a second search on the object's parent's DN to obtain the entire entry making provider search based on address inefficient.

  - HPD024 Resolution (4/26/10): First Option

395
  - CSD011 Issue: Will requiring inheritance of addresses (see CSD008) impair performance? If yes, is it "worth it" in order to improve manageability? Resolution (13-04-30): "class association" (UML modeling) of a native relational data design will address this issue.

- CSD012 (HPD025):

400
  - HPD025 Issue: For "memberof" query, ex: find the organizations that this provider is a memberof. Who is responsible for defining how many "memberof" relationship levels should be searched, and returned. 1) Should the HPD always search and return all levels, 2) Should the HPD only search one level at a time, and have the Consumer request subsequent searches, 3) Should the consumer identify "up to" how many levels the HPD should search 4)Should the consumer identify "up to" what type of organization to search

405
    to.

- HPD025 Resolution (4/28/10): This is determined by the LDAP standards and will not be defined in this profile.

- CSD012 Issue: CSD imposes a member-of rigor regarding Organization > Facility > FacilityService and, potentially, the Provider member-of relationships to Facility and/or (or just or?) Facility and FacilityService. Does this require us to revisit HPD025? Resolution (13-04-30): Leverage the existing information models; recast these information models as an XSD and employ XQuery against this XSD as the query transaction language. NOTE: a number of candidates were considered including: SPARQL, SQL, OCL, LDAP and XQuery. Previous issues with SQL caused this to appear to be a high risk option. SPARQL and OCL both have very low adoption in the marketplace. LDAP is problematic as a relational query approach for the reasons already outlined.

- CSD014:

  - Is it appropriate to constrain Services to be provided only at Facilities? There are cases where Organizations provide Services in the field (not thru facilities). Should there be a Service-Organization relationship and this *may* be provided at a Facility?

  - Conversely, where a Service-Facility relationship exists, does a relationship to the Organization need to be explicit or can it be inherited through the Facility-Organization relationship? Basically – can a single Facility have more than Organization relationship? If yes, the Facility-Service-Organization relationship needs to be explicit; if not, then where multiple Organizations are operating out of a single Facility there will need to be a contrivance of multiple Facility IDs (which could be confusing). Resolution: a relationship will be modeled between FacilityService and Organization, independent of the relationship between Facility and Organization. This will be an optional relationship which *may be* leveraged to clarify the auspices under which specific services are being provided.

- CSD015:

  - Should geocode be part of Address (preferred) or should it be a separate attribute, such as Position? Putting geocode within address, as a required element, puts a burden on existing deployments that may not have geocoded their addresses and it also is superfluous in cases such as "mailing" address (which may be PO box). Not requiring geocodes; however, undermines the ability to do any proximity calculations, which are key to the CSD value proposition. Addressed (13-05-14): geocode has been included as a separate attribute.

- **NOTE: the type of geocode (standard) has been specified… it is not left up to implementers.**

- CSD016:

  - Experience from HPD implementers is negative regarding the ability of LDAP to support relationships. This will be compounded for CSD, as relationships are the key element of

445   the CSD Profile's value proposition. A relational model is to be favored, and to query this, a format such as SQL or XQuery will work much better than LDAP as the query language. Of these, XQuery will likely better fit with the returning of xml or JSON responses. Resolution: embrace XQuery for the CSD Profile.

- CSD017:

450
  - It will simplify things to make the Provider-Organization relationship mandatory; e.g., a Provider who is in private practice would at least have that relationship to a "private practice" organization entity where the administrative information may be maintained. There is a downside in that this could be burdensome (from an IT standpoint) for the many GPs who are in private practice (lots of data maintenance). Is this too much

455   downside? Resolution: the Provider-Organization relationship will be mandatory.

- CSD018:

  - There will be attributes which should be associated with Provider-Organization, Provider-Facility and Provider-FacilityService. If class associations are employed, three new "classes" will be created, one for each of these relations. This is good design, but can

460   create a proliferation of classes. Proliferation is a concern as it starts to introduce complexity of the profile. **Should some functionality be shed in the interest of simplicity or does that just force implementers into workarounds? Decision to include attributes based on combinations.**

- CSD021

465
  - Each national or regional implementation will have differing facility "attributes" which MUST be maintained. Should this be modeled explicitly or should it be left up to each implementer to extend the data model as needed? The present draft of CSD explicitly models a facility "attribute" list as a mandatory set of coded-attribute plus value pairs. The jurisdiction would specify WHAT attributes are mandatory (e.g., number of beds,

470   catchment area, water point availability, etc.). Is such a design readily conformance-testable? What are the implications this has regarding XQuery queries (does this make it overly complicated or unwieldy)? **Does the proposed XSD format support a managed/interoperable way to support jurisdictions' definition of mandatory attributes which extend the current XSD? Or should an xs:any be used instead?**

475   **Decision: a managed extension mechanism was chosen based on the xs:any capability.**

- CSD022

  - The plain-text iCalendar format is much more widely adopted than the WS-Calendar or the xCal xml formats. This draft assumes plain-text iCalendar results will be returned

480   within a text element in a CSD query result. These results will be nested within the XQuery result underneath the provider or FacilityService entity to which they are related. Should the xCal or WS-Calendar standards be preferred because they are natively xml-based? **Resolution (13-05-14): favor the plain text version.**

- CSD023

485
  - Is there an existing standard for how XQuery queries are conveyed over the wire? There seem to be many ways to do this (some SOAP based, some REST based) but there does NOT seem to be a definitive spec on it. For now, the XQuery expression is passed inside an XML document which is POSTed to the InfoManager. There are potential challenges with this, including how results may be cached by the Consumer. **Is this HTTP/S POST**

490 **Option to be preferred? Should HTTPS be considered for communication between the consumer and the manager and TLS between the manager and the directory?** The security levels do not need to be the same for all the transactions, especially since it is possible that the consumer transactions will actually be from "consumers" who will not be able to do TLS. **Decision: use HTTP POST of an XML document specifying the**

495 **query.**

- CSD024

  - The mandatory requirement for facilityLanguage and facilityOperatingHours has been relaxed from R to R2. This reflects helpful feedback from the openHIE facility registry community regarding how rarely (at present) this information is known in developing

500 country settings where facility registries are currently being deployed. The change from R to R2 means these are now "minOccurs = 0" in the CSD.XSD and will potentially break use cases regarding discovering the language and discovering the operating hours (which has eReferral implications as well). There is an open question regarding whether it may be better to have a mandatory requirement for information that can have

505 "unknown" as its value or whether it is better to leave the data element off and have to figure out what to do on the client side when such data is missing. **Basically – should an XQuery against related elements of the data model return an "inner join" or an "outer join"? Decision: return an inner join.**

- CSD025

510
  - The unique ID format will be the II format from HL7[1] v3. The II data type will support cross referencing between multiple IDs; this will be supported in optional, repeating, otherID elements in the Organization, Facility, Service, and Provider directories. As described in the CSD.XSD, each "record" in each directory must have at least one globally unique ID and this is the one used for cross-referencing between the directory

515 data models. **Decision: use an OID**

- CSD026

  - Does the desire to support (optional) backward compatibility with HPD create a dependency which should be documented in Section 2.1? Resolution (13-06-03): Advice from the chair: "No".

520 - CSD027

---

[1] HL7 is the registered trademark of Health Level Seven International.

- W3C has defined a full set of error codes regarding XQuery; HTTP error codes are specified; CalDAV error codes are specified. **Do other error conditions need to be trapped and defined besides these?**

- CSD02*8*

525
- **Are ATNA log messages required for CSD queries?** If yes, what should be the nature of these? (Guidance from ITI chair: look for nature of the query, and who asked…). Decision: as with HPD, suggest but do not mandate ATNA.

- CSD029

530
  - ~~What wire protocol is best for conveying the Query for FreeBusy? The plan is to post an xml document containing the "text" format for vFREEBUSY. This does not seem ideal, however, as SOA web services calls are more common (e.g., Google Calendar).~~ Change (13-06-07): plan to use CalDAV (IETF RFC 4791) as this appears to have superseded the RFC 5545 specification. **It remains an open issue to prototype whether, unmodified, a service such as Google Calendar can act as a Service Availability Actor. Decision:**
535
    the is strong enough support to stick with CalDAV.

- CSD030

  - It will be a challenge to test generic XQuery at the Connectathon. Change (13-06-07): expect to employ XQuery user-defined functions to support "stored queries". **It remains an open issue to define/develop a set of stored queries (UDF XQueries) that satisfy**
540
    key use cases defined in part 1. Decision: support 4 "main" queries: organization, facility, provider, service.

- CSD032

  - Data consistency constraints are placed on Directory actors. **Is it necessary to provide a Service Finder with a mechanism to force the Care Services InfoManager to do a**
545
    refresh of its cached XML document? (NOTE: this could be accomplished by adding a Boolean "refresh" element as one of the choices in the CareServicesConsumerQuery element). Decision: this "force refresh" mechanism will not be made part of the profile.

- CSD033

550
  - A user defined function is still an XQuery and can be executed as one. **Is it useful to provide separate elements in the Find Matching Services [ITI-73] XML structure for free form XQuery vs. stored query invocation – even though both are, technically, XQuery queries? Decision: provide separate elements for functions (stored queries) and for expressions (ad hoc queries).**

555
- CSD035

  - **Should an InfoManager, as part of error handling, test for adherence to mandated code sets as well as testing for conformance to the CSD xsd? Decision: this test will not be included in the profile.**

- CSD036

560
- The InfoManager's error handling (page 58) stipulates that the InfoManager shall test for uniqueness of IDs. If duplicates are found, the Care Services InfoManager goes into a fault condition and does not respond to inbound queries from Consumers. **Is the described error handling sufficient? Decision: no more stringent error handling will be specified.**

565 **General Introduction**

Update the following Appendices to the General Introduction as indicated below. Note that these are not appendices to Volume 1.

Appendix A – Actor Summary Definitions

570 *Add the following actors to the IHE Technical Frameworks General Introduction list of Actors:*

Actor	Definition
Service Finder	The Service Finder Actor submits queries to the Care Services InfoManager, which returns the results in a reply document. Queries are expressed as parameterized stored query invocations or (optionally) as an ad hoc XQuery query. The Service Finder may optionally make queries against a Service Availability Actor as CalDAV vFREEBUSY queries.
Care Services InfoManager	The Care Services InfoManager Actor has two roles. The Care Services InfoManager maintains a local cache of information that represents cross-referenced data from one or more Care Services Directory actors. The Care Services InfoManager periodically refreshes this cache by querying Care Services Directory actors for updated content. The Care Services InfoManager's other role is to process inbound queries from Service Finder actors. These queries are executed against the cached content and the results are returned as a reply document.
Care Services Directory	The Care Services Directory Actor is responsible for returning an XML document in response to a request from a Care Services InfoManager. The response document contains all the content which has been inserted or updated in the Directory since the specified timestamp and is expressed in the format defined by the CSD Profile's XML schema.
Service Availability	The Service Availability Actor responds to requests for busy time for a provider at a service location, or for the service itself at a specified facility.

Appendix B – Transaction Summary Definitions

575 *Add the following transactions to the IHE Technical Frameworks General Introduction list of Transactions:*

Transaction	Definition
Find Matching Services [ITI-73]	The Find Matching Services transaction is used to express queries regarding locations, times, and kinds of services available at different facilities and from different providers and organizations. These are queries against the CSD schema for documents describing these services.
Query for Updated Services [ITI-74]	The Query for Updated Services is used to obtain all directory document content which has been inserted or updated since the specified timestamp. These are queries against the CSD schema for documents describing these services.
Care Services Free Busy Query [ITI-75]	The Query for FreeBusy is used to retrieve the busy scheduling information for a provider or service at a facility. It complies with the IETF CalDAV specification.

Glossary

Add the following glossary terms to the IHE Technical Frameworks General Introduction Glossary:

580   No new terms.

Volume 1 – Profiles

Copyright Licenses

Add the following to the IHE Technical Frameworks General Introduction Copyright section:

None

585 ## Domain-specific additions

1.7 History of Annual Changes

Add the following bullet to the end of the bullet list in Section 1.7

- Added the CSD Profile which supports a care services discovery service.

590

2.1 Dependencies among Integration Profiles

No modifications are needed to 2.1

Integration Profile	Depends on	Dependency Type	Purpose

595

Add Section 2.2.35

2.2.35 Care Services Discovery (CSD) Integration Profile

600 The CSD Profile supports queries across related directories containing data about: organizations, facilities, services and providers. Queries against an optional "FreeBusy" service are also supported; this FreeBusy information would enable the development of a list of schedulable time slots for providers or services at specific facilities.

Add Section 35

605

35 Care Services Discovery (CSD) Profile

The Care Services Discovery (CSD) Profile supports queries across related directories containing data about organizations, facilities, services and providers. Queries against an optional "FreeBusy" service are also supported; this FreeBusy information would enable the development of a list of schedulable time slots for providers or services at specific facilities.

The ability to query interrelated, federated, service directories supports important service discovery activities. This includes consumer-centric queries such as finding "where is the closest youth mental health services clinic" or "when is the earliest time I could schedule an appointment with a physiotherapist near my workplace". In addition, crucial health system management workflows are supported. This can include reporting and analyses, such as "what are my health human resource capacities, by facility, by cadre", "what are all the services offered at this facility" or, conversely, "where are all the facilities that offer this service". Lastly, the CSD Profile may be employed to support line-of-business queries such as "what is the enterprise facility ID for this facility" or "what is the encryption certificate and electronic endpoint for this provider". The CSD Profile may be employed to support, for example, the Provider Queries listed by the US Office of the National Coordinator as part of the Standards and Interoperability Framework (http://wiki.siframework.org/file/view/ESI%20Query%20and%20Response.pdf).

The CSD Profile describes four actors and the transactions between them:

1. **Service Finder** – the Service Finder Actor submits queries to the Care Services InfoManager, who returns the result(s) in a response document. Queries may be expressed by invoking "stored" queries or (optionally) by submitting well-formed ad hoc XQuery queries. Service Finder actors who support the FreeBusy Option may submit a FreeBusy query to a Service Availability Actor who will reply with a vFREEBUSY response.

2. **Care Services InfoManager** – the InfoManager Actor has two roles. The InfoManager maintains a local content cache that represents cross-referenced data from one or more Care Services Directory actors. The InfoManager periodically refreshes this cache by querying Directory actors for updated content. The InfoManager's other role is to process inbound queries from Service Finder actors. These queries are executed against the cached content. Results from an inbound query are returned to the Service Finder Actor in a response document.

3. **Care Services Directory** – the Directory Actor is responsible for returning an XML document in response to a refresh request from a Care Services InfoManager. The response document contains content which has been inserted or updated in the Directory since the last refresh and is expressed in the XML format defined by the CSD Profile's XML schema.

4. **Service Availability** – the Service Availability Actor responds to vFREEBUSY requests for busy time for a provider or service at a specified facility. The format of the request and response is defined by the IETF CalDAV specification (RFC 4791).

645   The CSD Profile's loosely coupled design and flexible querying capability means it can be deployed within a number of eHealth architectures and support a wide array of care workflows.

35.1 CSD Actors, Transactions, and Content Modules

Figure 35.1-1 shows the actors directly involved in the CSD Profile and the relevant transactions between them.

650
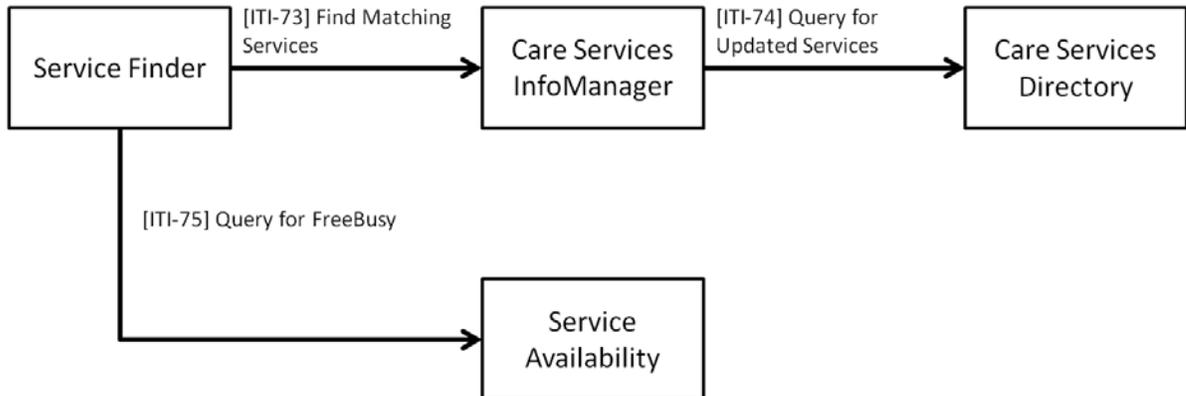
Figure 35.1-1: CSD Profile Actor Diagram

Table 35.1-1 lists the transactions for each CSD Profile Actor. To be conformant with the CSD
655   Profile, an implemented actor must perform all that actor's required transactions (labeled "R").
Transactions labeled "O" are optional.

Table 35.1-1: CSD Integration Profile - Actors and Transactions

Actors	Transactions	Optionality	TF Reference
Service Finder	Find Matching Services [ITI-73]	R	ITI TF-2c: 3.73
	Query for FreeBusy [ITI-75]	O	ITI TF-2c: 3.75
Care Services InfoManager	Find Matching Services [ITI-73]	R	ITI TF-2c: 3.73
	Query for Updated Services [ITI-74]	R	ITI TF-2c: 3.74
Care Services Directory	Query for Updated Services [ITI-74]	R	ITI TF-2c: 3.74
Service Availability	Query for FreeBusy [ITI-75]	R	ITI TF-2c: 3.75

35.1.1 Actor Descriptions and Actor Profile Requirements

660   ### 35.1.1.1 Care Services Directory

A Care Services Directory Actor shall maintain definitive data regarding one or more of:
Organization, Facility, Services, or Provider. A Care Services Directory shall be able to respond
to a Care Services InfoManager's request for new or updated content since a specified date and

665   time. To support this capability, a Care Services Directory shall support time stamped "journaling". In this way, for example, data elements that are deprecated are not simply deleted, but rather are updated to an appropriate status indicating their deprecation. In addition, a Care Services Directory shall also maintain the content needed to be able to construct and return an XML document conformant to the CSD XML schema definition.

670   In order for the Care Services InfoManager's cached content to be able to serve its role as an interlinked data source, the following conditions shall be met by Care Services Directory actors who maintain directory content.

1. Implementing jurisdictions may mandate code sets for Organization Type, Service Type, Facility Type, Facility Status, Provider Type, Provider Status, Contact Point Type, Credential Type, Specialization Code, and language code. Care Services Directory actors
675   shall be configurable to use these code sets, where mandated.

2. Implementing jurisdictions may mandate conventions regarding the types, components and formatting of Name, Address and AddressLine elements. Care Services Directory actors shall be configurable to use these formatting conventions, where mandated.

3. Implementing jurisdictions may mandate the source of truth regarding organization ID,
680   service ID, facility ID and provider ID. Care Services Directory actors shall ensure that all cross referenced IDs match corresponding directory records in the jurisdictionally mandated sources of truth.

For purposes of this profile, a "directory record" is defined as an `//organizationDirectory/organization`, `//serviceDirectory/service`,
685   `//facilityDirectory/facility` or `//providerDirectory/provider` element as defined in the CSD XML Schema Definition (ITI TF-2x: Appendix W).

A Services Directory shall support least one of the HTTP POST or SOAP protocols as defined Section ITI TF-2c: 3.74.4.2, for receiving Query for Updated Services [ITI-74] requests, and shall respond to each request using the received protocol.

690   **35.1.1.2 Service Finder**

The Service Finder Actor queries the Care Services InfoManager to retrieve interrelated content regarding care services. A Service Finder Actor invokes stored queries in order to execute the common care service discovery requests supported by the InfoManager. A Service Finder Actor shall be able to issue to the Care Services InfoManager one or more of the stored queries defined
695   in ITI TF-2c: 3.73.4.1.2.1 to 3.73.4.1.2.4. A Service Finder Actor should be able to invoke on the Care Services InfoManager other implementation-specific stored queries which the InfoManager supports.

A Service Finder Actor who supports the "Ad hoc XQuery" Option may submit a well-formed XQuery query to an InfoManager that supports the "Ad hoc XQuery" Option. Such XQuery
700   queries may return content in a format other than XML (e.g., JSON, CSV, etc.).

A Service Finder shall support at least one of the HTTP POST or SOAP protocols as defined Section ITI TF-2c: 3.73.4.1.4, for issuing Find Matching Services [ITI-73] requests.

A Service Finder Actor who supports the "FreeBusy" Option capability may execute a Query for FreeBusy query to a Service Availability Actor, who replies with a vFREEBUSY response for
705   the provider or service that was the subject of the query.

35.1.1.3 Care Services InfoManager

The Care Services InfoManager performs two functions:

1. On a periodic basis, the InfoManager refreshes its cached content with updated information from one or more Care Services Directory actors. The refresh period (e.g.,
710   every *x* minutes, hourly, daily, etc.) is established by the implementing jurisdiction and shall be configurable on the Care ServicesInfoManager.

2. When it receives an inbound query, the InfoManager executes the query against its cached content and returns a response document in reply. The inbound query may be either a stored query invocation or, if the InfoManager supports the Ad hoc XQuery
715   Option, a well-formed XQuery query.

A Care Services InfoManager shall be able to respond to all of the stored queries defined in ITI TF-2c: 3.73.4.1.2.1 to 3.73.4.1.2.4.

A Care Services InfoManager shall be able to query multiple Care Services Directory using [ITI-74].

720   A Care Services InfoManager shall support both the SOAP and HTTP POST transport protocols as defined in Section ITI TF-2c: 3.73.4.1.4 for receiving Find Matching Services [ITI-73] requests and shall respond to each request using the received protocol.

A Care Services InfoManager shall support both of the HTTP GET and SOAP protocols as defined Section ITI TF-2c: 3.73.4.1.4 for issuing Query for Updated Services [ITI-74] requests.
725   The protocol selected for a particular request could be selected, for example, by configuration.

35.1.1.4 Service Availability

The Service Availability Actor receives an inbound Query for FreeBusy request from a Service Finder that seeks "busy" information about a specified Provider or Service entity. The Service Availability Actor returns a "vFREEBUSY" response in reply.

730   ## 35.2 CSD Actor Options

Options that may be selected for each actor in this profile, if any, are listed in the Table 35.2-1. Dependencies between options when applicable are specified in notes.

Table 35.2-1: CSD - Actors and Options

Actor	Option Name	Reference
Service Finder	Ad hoc XQuery	Section 35.2.1
	FreeBusy	Section 35.2.2
Care Services InfoManager	Ad hoc XQuery	Section 35.2.1

Actor	Option Name	Reference
Care Services Directory	No options defined	- -
Service Availability	No options defined	- -

35.2.1 Ad hoc XQuery Option

A Service Finder Actor that supports the Ad hoc XQuery Option shall be able to expresses a well-formed ad hoc XQuery query against the CSD element of the XML schema for CSD. See ITI TF-2x: Appendix W. This query will conform to the XQuery specification referenced in ITI TF-2c: 3.73.3.

A Care Services InfoManager Actor that supports the Ad hoc XQuery Option shall be able to respond to ad hoc XQuery queries from a Service Finder Actor. See ITI TF-2c: 3.73.3.

35.2.2 FreeBusy Option

A Service Finder Actor that supports the FreeBusy Option shall implement the Query for FreeBusy [ITI-75] transaction. See ITI TF-2c: 3.75.

35.3 CSD Required Actor Groupings

CSD Actor groupings are listed in Table 35.3-1.

Table 35.3-1: Care Service Discovery- Required Actor Groupings

CSD Actor	Actor to be grouped with	Reference	Content Bindings Reference
Service Finder	None	-	-
Care Services InfoManager	None	-	-
Care Services Directory	None	-	-
Service Availability	None	-	-

35.4 CSD Overview

35.4.1 Concepts

The Care Services Discovery Profile supports queries against interrelated information regarding: organizations, facilities, providers and services. The interrelationship between these data is illustrated in Figure 35.4.1-1.
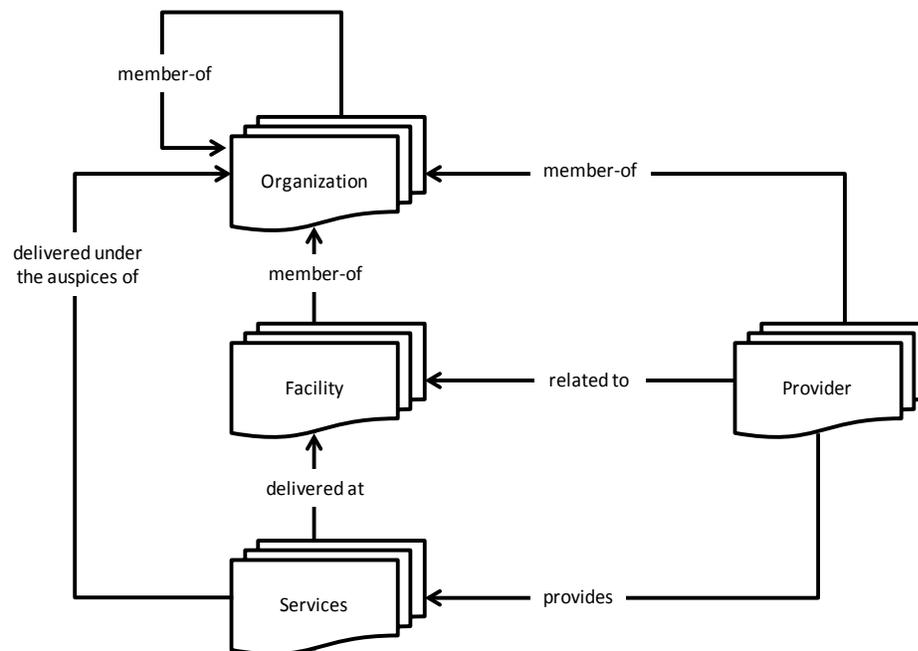
755   **Figure 35.4.1-1: Top-level Relationships between Care Services Directories**

Optional support is also provided to query for FreeBusy information about providers or services at specific facilities.

The following provides a high level description of the directories shown in Figure 35.4.1-1:

760   1. Organization – Organizations are "umbrella" entities; these may be considered the administrative bodies under whose auspices care services are provided such as Healthcare Information Exchanges (HIEs), Integrated Delivery Networks (IDNs), Non-government Organizations (NGOs), faith-based organizations (FBOs) or even a one-physician family practice. An organization has a unique identifier and administrative attributes such as

765   contact person, mailing address, etc. An organization may have a parent-child relationship with another organization.

    2. Facility – Facilities are physical care delivery sites such as hospitals, clinics, health outposts, physician offices, labs, pharmacies, etc. A Facility has a unique identifier, geographic attributes (address, geocode), contact attributes, attributes regarding its hours

770   of operation, etc. Each Facility is related to at least one Organization and to at least one Service. A Facility may be related to zero, one or more Providers.

    3. Service – Each care service has a unique identifier. Examples include: surgical services, antenatal care services or primary care services. The combination of a Service offered at a Facility may have specific attributes including contact person, hours of operation, etc.

775   4. Provider – A Provider is a health worker as defined by WHO; a provider might be a physician, nurse, pharmacist, community health worker, district health manager, etc. Providers have contact and demographic attributes. Each Provider is related to at least

one Organization and may be related to one or more facilities and services. Specific attributes may be associated with the Provider relationship with these other entities.

780 Because of the particular challenges regarding the representation of names across multiple languages and in different cultures, attention is drawn to the ITI TF-2a: 3.24.5.2.3.1. This section in the ITI Technical Framework describes the use of the language tag as documented in IETF RFC 1766 and the HL7 XCN name data type. The CSD XSD has expressly defined the commonName element so that implementers may make use of these methods in supporting 785 complex international conventions regarding names.

A top-level view of the CSD Profile's process flow is shown in Figure 35.4.1-2. Interrelated directory content is maintained by the Care Services InfoManager. The Care Services InfoManager routinely obtains new or updated content from the CSD Directories by polling them. These updates refresh a data cache which the InfoManager maintains. The InfoManager's 790 cache is refreshed at an appropriate interval specified by the implementing jurisdiction. Implementing jurisdiction will consider the implications of out of date information when setting the refresh interval between cache updates. The normal delays in updating directories will also be part of this consideration.

The Service Finder is the actor that queries for information about interrelated care services. 795 These queries are sent to the Care Services InfoManager who develops a response based on the content in its cached data store. To execute a content query, a Service Finder Actor invokes a stored query against the InfoManager's cached data. Where both actors support the Ad hoc XQuery Option, a Service Finder may optionally submit a well-formed XQuery to an InfoManager, who returns the response in a reply document.

800 In addition, a Service Finder that supports the FreeBusy Option may submit a FreeBusy query to a Service Availability Actor who returns the "busy" time for the Provider or Service in question. The Service Finder may net the busy time against the Provider's or Service's operating hours to calculate the "free" (schedulable) time slots which may be available.

805
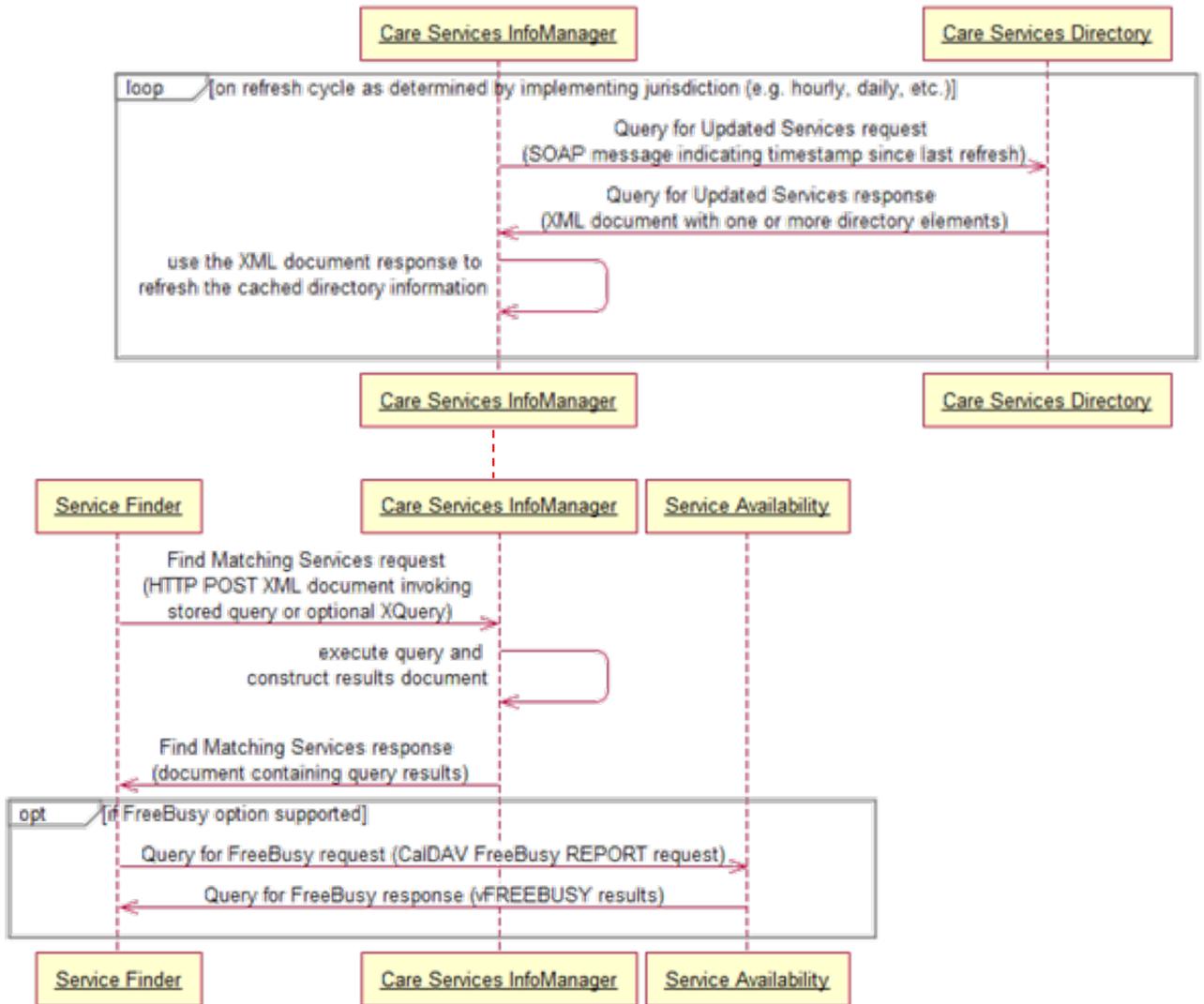
```
Diagram pseudocode:

participant "Care Services InfoManager"
participant "Care Services Directory"

loop on refresh cycle as determined by implementing jurisdiction (e.g., hourly, daily, etc.)
 Care Services InfoManager -> Care Services Directory: Query for Updated Services request \n(SOAP
message indicating timestamp since last refresh)
 Care Services Directory -> Care Services InfoManager: Query for Updated Services response \n(XML
document with one or more directory elements)
 Care Services InfoManager -> Care Services InfoManager: use the XML document response to
\nrefresh the cached directory information
end

======================================================================

participant "Service Finder"
participant "Care Services InfoManager"
participant "Service Availability"

Service Finder -> Care Services InfoManager: Find Matching Services request \n(HTTP POST XML
document invoking \nstored query or optional XQuery)
Care Services InfoManager -> Care Services InfoManager: execute query and \nconstruct results
document
Care Services InfoManager -> Service Finder: Find Matching Services response \n(document
containing query results)

opt if FreeBusy Option supported
 Service Finder -> Service Availability: Query for FreeBusy request (CalDAV FreeBusy REPORT
request)
 Service Availability -> Service Finder: Query for FreeBusy response (vFREEBUSY results)
end
```

Figure 35.4.1-2: Process Flow

35.4.2 Use Cases

There are two "exemplary" use cases for the CSD Profile: health human resource management and clinical provider appointment availability. These two use cases illustrate using CSD to query across related provider and facility data and using the profile to search for the soonest available appointment time slot for a specific type of clinical specialist.

35.4.2.1 Use Case #1 Health Worker Staff Levels Report

35.4.2.1.1 Health Worker Staff Levels Use Case Description

A district health (DH) manager wishes to generate a listing of the health worker staffing levels for each facility in his district. These data should be summarized by health worker type (e.g., community health worker, nurse assistant, nurse, physician, etc.).

35.4.2.1.2 Health Worker Staff Levels Report Process Flow

- The district health information system, acting as a Service Finder, invokes an implementation-specific stored query on the Care Services InfoManager. It passes the appropriate query parameters indicating the district of interest and other data.

855
- The Care Services InfoManager executes the stored query and returns the results to the district health information system. The district health information system presents the query results to the DH manager for analysis and/or reporting.

The interactions between the various actors in this use case are shown in Figure 35.4.2.1.2-1.
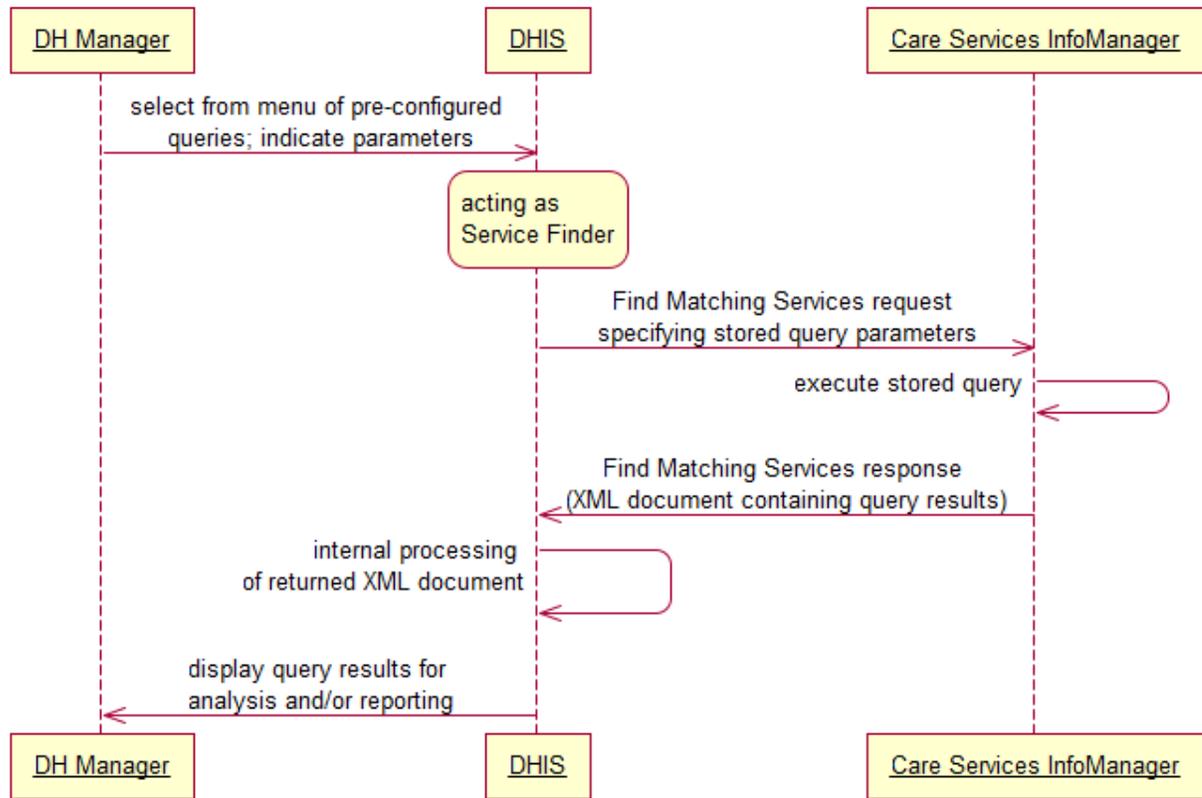

```
Diagram pseudocode:

participant "DH Manager"
participant DHIS
participant "Care Services InfoManager"

DH Manager -> DHIS: select from menu of pre-configured \nqueries; indicate parameters

state over DHIS: acting as \nService Finder

DHIS -> Care Services InfoManager: Find Matching Services request \nspecifying stored query
parameters
Care Services InfoManager -> Care Services InfoManager: execute stored query
Care Services InfoManager -> DHIS: Find Matching Services response \n(XML document containing
query results)
DHIS -> DHIS: internal processing \nof returned XML document
DHIS -> DH Manager: display query results for \nanalysis and/or reporting
```

860

865

870

875

Figure 35.4.2.1.2-1: Health Worker Staff Levels Use Case

35.4.2.2 Use Case #2: Provider Availability Query

35.4.2.2.1 Provider Availability Query Use Case Description

880 The patient, David Lambert, consults with his physician who recommends surgery. The patient needs to find a suitable surgeon, taking into consideration the location and availability of the surgeons.

35.4.2.2.2 Provider Availability Query Process Flow

- David Lambert sees his family physician, Dr. Black, regarding a recent knee injury.

885 - Dr. Black diagnoses the problem as a torn ACL and decides to refer David to an orthopedic surgeon.

- Dr. Black uses his EMR query tool, which implements a Service Finder Actor, to search for orthopedic surgeons within 30km of David's home who have available consult timeslots in the next 30 days.

890 - The EMR retrieves the information and display it to Dr. Black.

- David and Dr. Black decide on an orthopedic surgeon; Dr. Black prepares an eReferral.

The interactions between the actors are shown in Figure 35.4.2.2.2-1.
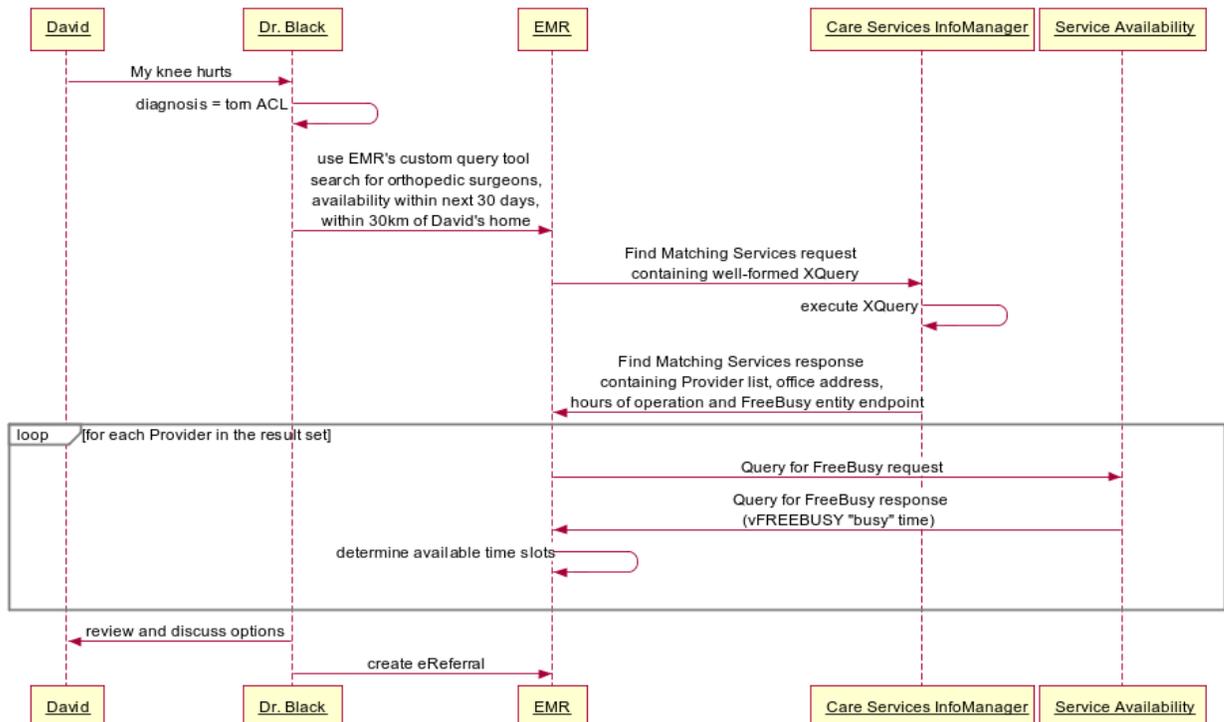
```
Diagram pseudocode:

participant David
participant "Dr. Black"
participant EMR
participant "Care Services InfoManager"
participant "Service Availability"

David -> Dr. Black: My knee hurts
Dr. Black -> Dr. Black: diagnosis = torn ACL
Dr. Black -> EMR: use EMR's custom query tool \nsearch for orthopedic surgeons, \navailability
within next 30 days, \nwithin 30km of David's home

state over EMR: acting as: \nService Finder

EMR -> Care Services InfoManager: Find Matching Services request \ncontaining well-formed XQuery
Care Services InfoManager -> Care Services InfoManager: execute XQuery
Care Services InfoManager -> EMR: Find Matching Services response \ncontaining Provider list,
office address, \nhours of operation and FreeBusy entity endpoint

loop for each Provider in the result set
 EMR -> Service Availability: Query for FreeBusy request
 Service Availability -> EMR: Query for FreeBusy response \n(vFREEBUSY "busy" time)
 EMR -> EMR: determine available time slots
end

Dr. Black -> David: review and discuss options
Dr. Black -> EMR: create eReferral
```

Figure 35.4.2.2-1: Provider Availability Use Case

35.4.2.3 Other Provider Information Query Use cases (from the IHE HPD Profile):

1. **Provider Authorization and lookup during an emergency event:** During Hurricane Katrina, health care volunteers were turned away from disaster sites because there was no means available to verify their credentials. At an emergency site, the CSD service can be queried to quickly identify and grant permission to credentialed providers to enter the scene.

2. **Keeping agency provider information current:** A German government agency dealing with healthcare services for its constituents wishes to keep its agency's healthcare provider information current. The agency determines that it will use the CSD Profile to access the most current provider information from a Care Services InfoManager. On a regular basis, the agency queries the Care Services InfoManager to provide a list of the updated information.

3. **Providing Personal Health records to a new Primary Care Physician:** An individual has changed health plans. As a result that individual must change his Primary Care Physician. The individual has a Personal Health Record and would like to provide that information to his new Primary Care Physician. The individual needs to determine where to have the Personal Health Record transmitted to.

4. **Certificate Retrieval:** National regulations in many European countries require that an electronically transmitted doctor's letter is encrypted in a way that only the identified

receiver is able to decrypt it. In order to encrypt the letter, the sender has to discover the encryption certificate of the receiver.

945   5. **Language Retrieval:** An individual who only speaks Italian requires healthcare services at an Outpatient Clinic. That individual would like to be able to communicate with the Clinic personnel in his native language. The individual or his caregiver needs to determine which clinic supports Italian and provides the care service that is required.

35.4.2.4 Specific other use cases supported/assisted by the CSD Profile:

950   The following list represents end-user requirements to be supported by the CSD Profile or supported by applications, such as GIS applications, assisted by the profile. This list was developed based on meetings with Ministry of Health officials in Tanzania:

1. Generate a full list of facilities, including GPS coordinates, sorted by region or district. Filter the list by specific Service or Services offered at the facility.

955   2. Search for facilities by facility name; view details about each facility (operational status, contact information, type), search by district, display the facility information on a map based on the GPS coordinates.

3. Quantify facilities based on service levels and types, locations based on types and service levels, understand resources for a given facility (ICT applications, beds, etc.), catchment
960   area viewing to determine levels of service provision for a geographic area, contextual metadata about facility types inline with view of facility attributes, potentially track other facilities outside of health facilities, filter map views by ownership type (to help look for collaborative opportunities).

4. Filter facility by implementing partner, filter by service and region, view data for filtered
965   list one by one both as viewing one per site, and in context of all (map view of facilities based on their GPS coordinates), available mobile networks for facility, power source.

5. Filter facilities by service, admin hierarchy, view in web application, along with facility details, , subservices provided by facilities (i.e., kangaroo care within a facility)

The following list represents two key use cases articulated by the Rwanda Health Enterprise
970   Architecture (RHEA) Provider Registry project (shown here, unmodified from the project documentation):

1. This query takes in a unique ID for a provider and returns the enterprise ID associated with the provider who matches the given ID.

975

"Primary Actor"	Interoperability Layer, potentially Point Of Care Application
Pre-Conditions	The application must be authenticated to the system
Success Guarantee	An enterprise id (EID) is returned to the requesting application
Main Success Scenario	1. The "actor" submits a request with a given ID type and ID number 2. The system returns the EID
Extensions	2.a The system does not find the provider and returns an error code.

980   2. Queries for HC Professional using a variety of possible parameters. This transaction returns a list of all professionals that match the criteria specified in the query parameters. Otherwise, if an error occurred, then an error is returned.

"Primary Actor"	Interoperability Layer, Point Of Care Application?
Pre-Conditions	The application must be authenticated to the system
Success Guarantee	An list of enterprise ids (EIDs) is returned to the requesting application
Main Success Scenario	1. The "actor" submits a request with a given query parameters 2. The system returns a list of matching EIDs
Extensions	2.a The system does not find the provider and returns an error code. 2.b The requested search parameters are invalid and return an error code.

35.5 CSD Security Considerations

985   The CSD Profile does not support the exchange of person-centric health information. Therefore, this profile does not specify security mechanisms, such as the Audit Trail and Node Authentication (ATNA) Profile, that would be required were that the case. However extension points have been provided which would allow for support of one or more security and privacy profiles developed by IHE or other entities.

990   Information maintained by Care Services Directory actors may be subject to privacy policies and mandates. Users of this profile should refer to relevant privacy legislation, codes of fair information practice and other guidelines so as not to breach personal privacy in their collection, use, storage and disclosure of provider information (for example), including any consent requirements. Privacy legislation may require users to consider their particular set of

995   circumstances (i.e., location and sector) and whether privacy legislation covers those circumstances.

Provisions in health professional legislation and regulations shall also be considered when unique provider identifiers and any subsequent identifiable information is collected, stored and shared. If there is any information, in part, or whole, as defined by the CSD Profile schema or any extensions to the given schema, that is considered private or sensitive, then appropriate

1000   security and privacy procedures shall be considered to protect that information.

This section provides an overview of the typical risks that should be mitigated if desired, and recommends the corresponding IHE profile(s) that should be combined with CSD.

For general IHE risks and threats, see ITI TF-1: Appendix G.

1005   Some categories of risks currently identified for consideration are:

- The possibility of inaccurate or unverified data being maintained in a Care Services Directory.

- Accidental release of information protected by additional regulation or law.

- Malicious attacks to the data, modifying the data either in transmission or at the Care
1010   Services InfoManager or Service Availability actors.

- Policies and procedures for the verification, validation, and reconciliation of data are ineffective or inconsistent across authoritative Care Services Directory actors.

- Any activity performed on CSD data by a Service Finder Actor with the expectation that data are authoritative. For instance, PHI could be transmitted to a FAX number, a mailing
1015   address, or a health internet address with the expectation that the addresses are authoritative.

The following shall be considered by implementers of the profile, as part of the planning process for implementation:

- Implement and enforce policies and procedure to validate the provider information before
1020   it is stored and published by a Care Services Directory.

- The Audit Trail and Node Authentication (ATNA) Profile should be considered to ensure that nodes in a network are authenticated.

- The ATNA Profile may be employed to capture and record audit trail events related to the Query for Updated Services transaction.

1025   - The Enterprise User Authentication (EUA) Profile may be employed to authenticate Service Finder and/or Care Services InfoManager actors if access to query information is to be limited and controlled. If CSD content is going to be public, open and searchable by anyone, the user authentication may not be applicable.

- The Cross-Enterprise User Assertion (XUA) Profile may be employed to assert Service
1030   Finder and/or Care Services InfoManager credentials for the purpose of enforcing Role

Based Access Control (RBAC) and protecting any non-public sensitive information if access to query information is limited and controlled.

- The Personnel White Pages (PWP) Profile provides a repository that may be used to hold system users' identification data.

1035
- Best practices should be employed regarding operations management for network monitoring, intrusion detection and secured data backups.

Implementers may follow these IHE profiles to fulfill some of their security needs. It is understood that institutions must implement policy and workflow steps to satisfy enterprise needs and to comply with regulatory requirements.

1040 The CSD implementer is further advised that many risks associated with physical security and operational management cannot be mitigated by the CSD Profile and instead responsibility for mitigation is transferred to the CSD implementer.

Specific Security Considerations Regarding the Use of XQuery

Because the CSD Profile relies on XQuery for ad hoc queries, specific care should be taken to
1045 ensure the safety and security of queries and content, including with regard to "injection" attacks (similar to the more widely found SQL injection attacks). One protection regarding the defense of privacy of XML content exchanged between actors is to require the use of encrypted transport. To satisfy such a requirement, Find Matching Services [ITI-73] and Query for Updated Services [ITI-74] transactions may be conducted using HTTPS.

1050 In addition, XQuery injection attacks may be guarded against using the following techniques (from http://www.balisage.net/Proceedings/vol7/html/Vlist02/BalisageVol7-Vlist02.html):

> *The most common way to block these kinds of attacks is to "escape" the dangerous characters or "sanitize" user inputs before sending them to the XQuery engine.*
>
> *In an XQuery string literal, the "dangerous" characters are:*

1055
> - *The & that can be used to make references to predefined or digital entities and needs to be replaced by the &*
>
> - *The quote (either simple or double) that you use to delimit the literal that needs to be replaced by ' or "*

35.6 CSD Cross Profile Considerations

1060 None

Appendices

None

Volume 2c – Transactions

1065

3.73 Find Matching Services [ITI-73]

3.73.1 Scope

1070 The Find Matching Services transaction is used to express queries against the CSD schema. This schema is found in ITI TF-2x: Appendix W. The query against the CSD schema may be expressed as an invocation of a stored query or (optionally) as a well-formed ad hoc XQuery.

The query is submitted by the Service Finder to the Care Services InfoManager by constructing an XML document expressing the query and executing an HTTP POST transaction. The Care Services InfoManager executes the query and returns the results in a response document.

1075 ### 3.73.2 Actor Roles
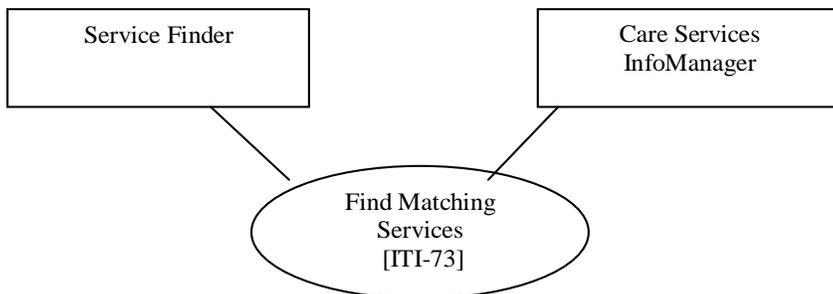
Figure 3.73.2-1: Basic Process Flow for Find Matching Services

1080 **Table 3.73.2-1: Actor Roles**

Actor:	Service Finder
Role:	Constructs and submits to the Care Services InfoManager an XML document expressing a stored function invocation or (optionally) a well-formed Ad hoc XQuery. The result from the Care Services InfoManager is returned in a document.

Actor:	Care Services InfoManager
Role:	The Care Services InfoManager receives from the Service Finder an XML document containing a stored query invocation or (optionally) a well-formed Ad hoc XQuery. The Care Services InfoManager processes the inbound query and returns the result to the Service Finder in a document.

3.73.3 Referenced Standards

The Find Matching Services [ITI-73] transaction relies on the following standards:

- The format of the Find Matching Services [ITI-73] XQuery or stored function invocation shall adhere to the W3C specification: **XQuery 1.0: An XML Query Language (Second Edition)** found at: http://www.w3.org/TR/xquery/. Stored functions exposed and supported by a Care Services InfoManager to Service Finders shall have been defined as XQuery User Defined Functions.

- By its inclusion in the XQuery standard, this means the XML Schema standard is also referenced; the current version of which can be found here: http://www.w3.org/TR/xmlschema11-1/.

- Queries are conveyed by the Service Finder to the Care Services InfoManager using one of:

    - HTTP. The W3C HTTP 1.1 standard is specified here: http://www.w3.org/Protocols/.

    - Simple Object Access Protocol (SOAP version 1.2) as specified by the W3C (http://www.w3.org/TR/soap/) and as profiled by IHE in ITI TF-2x: Appendix V

- The data model and transport protocols for the parameters for stored queries are defined according to the XForms 1.1 http://www.w3.org/TR/xforms/ in particular the `instance`, `submission`, and `bind` elements as described in the XForms Core Model of http://www.w3.org/TR/xforms/#structure-abstract.

- The EXPath Candidate Module 9 January 2010 HTTP Client Model http://expath.org/spec/http-client.

- The CSD data schema is based on the following content specifications:

    - IHE Health Provider Directory (HPD) Profile (and by reference ISO 21091)

    - WHO draft Master Facility List guideline

    - HL7 OMG Health Services Directory (HSD) Service Functional Model

    - HL7 Fast Health Interoperability Resources (FHIR®[2]) model

---

[2] FHIR is the registered trademark of Health Level Seven International.

3.73.4 Interaction Diagram
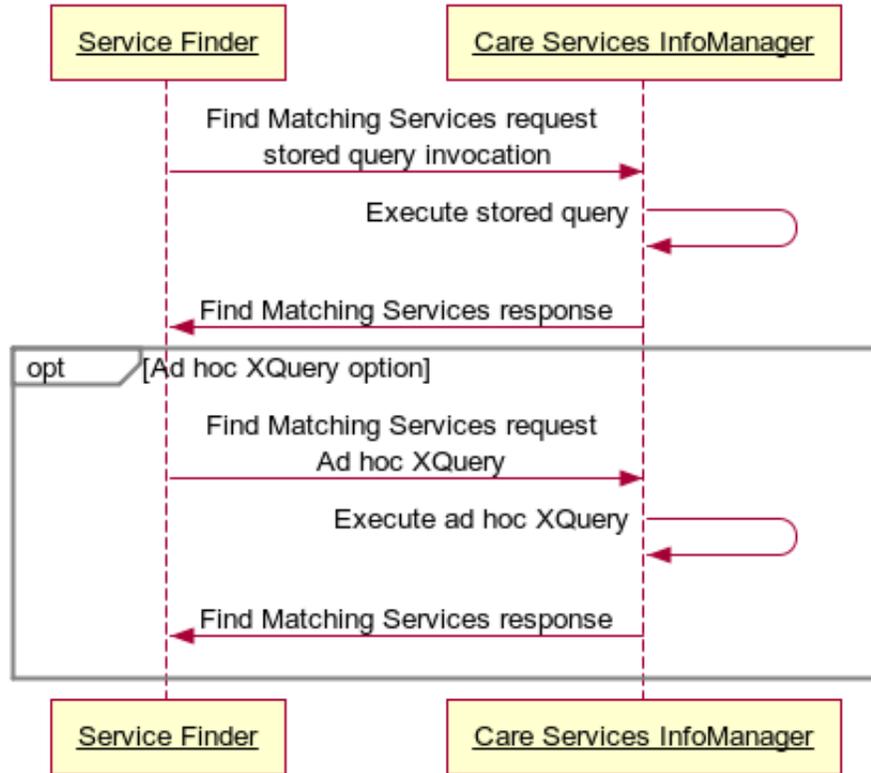
1110

```
Diagram pseudocode:

participant "Service Finder"
participant "Care Services InfoManager"

Service Finder -> Care Services InfoManager: Find Matching Services request \nstored query
invocation
Care Services InfoManager -> Care Services InfoManager: Execute stored query
Care Services InfoManager -> Service Finder: Find Matching Services response

opt Ad hoc XQuery Option
 Service Finder -> Care Services InfoManager: Find Matching Services request \n Ad hoc XQuery
 Care Services InfoManager -> Care Services InfoManager: Execute ad hoc XQuery
 Care Services InfoManager -> Service Finder: Find Matching Services response
end
```

1115

1120

1125

Figure 3.73.4-1: Find Matching Services [ITI-73] Interactions

3.73.4.1 Find Matching Services Request message (Stored Query)

In the definition of the Find Matching Services Request the following XML namespaces are
1130   used.

Table 3.73.4.1-1: Namespace Definitions

csd	urn:ihe:iti:csd:2013

The csd namespace is also as used as the default element namespace for describing elements of "CSD" and "Find Matching Services Request" documents.

3.73.4.1.1 Trigger Events

1135 The Service Finder sends a Find Services Request when it needs to access service information maintained by the Care Services InfoManager and for which the InfoManager has defined a Stored Query, as defined in Section 3.73.4.1.1.1.

The Care Services InfoManager shall have a Stored Query associated with each of the URNs referenced in the Required Stored Queries in Section 3.73.4.1.2 in Sections 3.73.4.1.2.1 through 1140 3.73.4.1.2.4.

In addition, the Care Services InfoManager may optionally make available other Stored Queries. The transaction for "publishing" such optional stored queries to the Care Services InfoManager is not defined by IHE.

3.73.4.1.1.1 Stored Queries

1145 The Query for Updated Services [ITI-74] and the Find Matching Services [ITI-73] transactions are defined as the invocation of a *Stored Query*. A jurisdiction may specify additional functionality through the definition of additional Stored Queries. A Stored Query is defined by a `csd:careServicesFunction` element. A `csd:careServicesFunction` is subject to the following conditons:

1150
- `csd:careServicesFunction` shall have an `@urn` attribute with a unique URN according to RFC 2141 and which has no reserved characters as defined in Section 23 of RFC 2141.

- `csd:careServicesFunction` shall contain an XFORM `model` element to define the syntax of the message for a particular transaction. The `@id` attribute of the `model`
1155   element shall be the same as the `@urn` attribute of the containing `csd:careServicesFunction`. The message semantics are further defined in Section 3.73.4.1.2.

- The `{$LOCATION}` shall be replaced with the URI of the base service address for the actor receiving actor.

1160
- The text content of the `csd:careServicesFunction/csd:definition` element must be XQuery MainModule content as defined in:

    http://www.w3.org/TR/xquery/#prod-xquery-MainModule

The semantic definitions of the query parameters for the transactions and the associated expectations on the actors are defined:

1165
- in Section 3.73.4.3.2 for a Care Services Finder initiating the Find Matching Services [ITI-73] (Ad hoc XQuery option) transaction against a Care Services InfoManager.

- by the jurisdiction for a Care Services Finder initiating a Find Matching Services [ITI-73] transaction with an optional Stored Query as described in Section 3.73.4.1.1.1 against a Care Services InfoManager.

1170

3.73.4.1.2 Message Semantics

As indicated in Figure 3.73.4.1.2-1, the `function` element has a mandatory attribute `@urn`. This attribute is used to specify the stored query ID. Each of the stored queries defined in Sections 3.73.4.1.2.1 to 3.73.4.1.2.4 has been assigned a unique ID.

1175 The message semantics for a Find Matching Services request shall be a Stored Query. This transaction defines four Stored Queries. The query parameters of these four Stored Queries are defined (formatted as per the XFORMS core model specification) as the content of a `csd:requestParams` element according to the following:

- Organization Search – see Section 3.73.4.1.2.1

1180
- Provider Search – see Section 3.73.4.1.2.2

- Facility Search – see Section 3.73.4.1.2.3

- Services Search – see Section 3.73.4.1.2.4

A Care Services InfoManager shall support all of the Stored Queries. A Service Finder shall support at least one.

1185 The `csd:careServicesFunction` XML required to define the four Stored Queries is found in ITI TF-2x: Appendix Y, Figure Y.3-1 through Y.3-4. For a non-normative XQuery implementation, see ITI TF-2x: Appendix W.

The Service Finder shall convey the Find Matching Services request using either the HTTP POST or SOAP transport protocols as defined in Section 3.73.4.1.4.

1190 Example content of a SOAP query request is shown in Figure 3.72.4.1.2-1 below. The schema for the `csd:careServicesRequest` element is contained in ITI TF-2x: Appendix Y.2.

An example query request is shown below.

1195
1200
```
<csd:careServicesRequest xmlns:csd="urn:ihe:iti:csd:2013" >
    <csd:function urn='urn:ihe:iti:csd:2014:stored-function:facility-search'>
      <csd:requestParams>
        <csd:codedType code='OPC-example' codingSchema="USDVA-example"/>
        <csd:address>
            <csd:addressLine component='city'>Chapel Hill</csd:addressLine>
         </csd:address>
        <csd:max>5</csd:max>
      </csd:requestParams>
    </csd:function>
</csd:careServicesRequest>
```

1205 **Figure 3.73.4.1.2-1: Example Request**

A Service Finder Actor shall be able to issue to the Care Services InfoManager one or more of the stored queries defined in Sections 3.73.4.1.2.1 to 3.73.4.1.2.4.

1210 For all of the stored queries defined in Sections 3.73.4.1.2.1 to 3.73.4.1.2.4, rudimentary pagination is supported through the use of the optional `start` and `max` parameters. The behaviors supported by these two parameters are as follows.

The Service Finder may optionally specify the `start` parameter to indicate the starting ordinal of the resulting query response. By default, the Care Services InfoManager begins the query response document at the first record of the query result set. To request that the response begin at 1215 record 51, however, the Service Finder would specify parameter `<start>51</start>`.

The Service Finder may optionally specify the `max` parameter to denote the maximum number of records the Care Services InfoManager returns in a query response. By default, all records are returned in a query response. To request that the Care Services InfoManager return 50 records in the response, the Service Finder would specify parameter `<max>50</max>`.

1220 A Service Finder Actor with limited data processing capability could choose to execute multiple queries with the same search parameters, but each with appropriately defined `start` and `max`, to support a paging behavior within its application. NOTE: if the cached data at the Care Services InfoManager has changed between subsequent requests the returned results shall reflect these changes.

1225 **3.73.4.1.2.1 Organization Search**

The careServicesFunction XML required to define the Organization Search careServicesFunction can be found in ITI TF-2x: Appendix Y, Figure Y.3-1. For a non-normative XQuery implementation of the organization search, see ITI TF-2x: Appendix W.

Description

1230 Performs a search for all organizations by name, type, address or ID.

The result set consists of all organizations matching the *search parameters* (id, `primaryName`, `name`, `codedType` and `address`).

The results set may be further restricted according to the *limit parameters* (`start`, `max`, `record/@status` and `record/@updated`). An ordering of the result set is not specified.

1235 Content-Type: text/xml
URN: urn:ihe:iti:csd:2014:stored-function:organization-search

Response

Results are returned as a valid CSD document with a root document element of `CSD`. The results set is contained entirely within the `organizationDirectory` element and consists of the full 1240 content of the `organization` elements of matching organizations as maintained by the Care Services InfoManager.

Parameters

Query Parameters are defined (formatted as per the XFORMS core model specification) as the content of following elements relative to the `requestParams` child element of `function`:

1245
- `id` *csd:uniqueID*: (optional) If present and the `@entityID` attribute contains a non-empty value, it is an ID which uniquely identifies an organization. This is an exact match.

- `otherID` *csd:otherID*: (optional) If present and the `@assigningAuthorityName` attribute contains a non-empty value, then the result set is restricted to only those organizations which have a `otherID` with the given `@assigningAuthorityName` and 1250 `@code`

- `primaryName` *xs:string*: (optional) If present and contains a non-empty value, then the result set shall be restricted to those organizations whose `primaryName` contains this value. Case insensitive.

- `name` *xs:string*: (optional) If present and contains a non-empty value, then the result set 1255 shall be restricted to those organizations which have a `primaryName` or `otherName` element containing this value. Case insensitive.

- `codedType` *csd:codedtype*: If present and contains a non-empty value the result set shall be restricted to those organizations whose `codedType/@code` equals this value for the coding schema specified by the `@codingSchema` attribute. Case insensitive.

1260
- `address` *csd:address*: (optional) Contains any number of child `addressLine` elements as follows:

  - Text content *xs:string*: (optional) If present and contains a non-empty value, then the results set shall be restricted to those organizations whose have an `addressLine` with specified `@component` containing this value exactly. Case insensitive.

1265
  - `@component` *xs:string* : (Required attribute) The component of the address we are searching. Case insensitive.

- `parent` *csd:organization*: (optional) The parent organization of the organization. If present and has an `@entityID` attribute, then the result set shall be restricted to those organizations whose parent organization is the one referenced by `@entityID`.

1270
- `start` *xs:int*: (optional) The starting index for results returned. Defaults to 1, which indexes the first organization matching the search parameters

- `max` *xs:int*: (optional) The maximum number of results returned. A value of less than zero implies no maximum.

- `record` *csd:record* : (optional) A child element to limit results according to

1275
  - `@status` *xs:string*: (optional) If present and contains a non-empty value, the result set shall be restricted to those organizations whose `record/@status` equals this value. Case insensitive.

- @updated *xs:dateTime*: (optional) If present and contains a non-empty value, the result set shall be restricted to those organizations whose record/@updated is at least the given value.

1280

```
<careServicesRequest>
    <function urn='urn:ihe:iti:csd:2014:stored-function:organization-search'>
      <requestParams>
        <codedType codingSchema="moh.gov.rw" code="FBO" />
        <address>
           <addressLine component='city'>Kigali</addressLine>
         </address>
         <max>5</max>
      </requestParams>
    </function>
</careServicesRequest>
```

1285

1290

Figure 3.73.4.1.2.1-1: Example careServicesRequest

1295 **3.73.4.1.2.2 Provider Search**

The careServicesFunction XML required to define the Provider Search careServicesFunction can be found in ITI TF-2x: Appendix Y, Figure Y.3-2. For a non-normative XQuery implementation of the provider search, see ITI TF-2x: Appendix W.

Description

1300 Performs a search for all providers by name, type, address or ID.

The result set consists of all providers matching the *search parameters* (id, otherID, commonName, codedType and address).

The results set may be further restricted according to the *limit parameters* (start, max, record/@status and record/@updated). An ordering of the result set is not specified.

1305 Content-Type: text/xml
URN: urn:ihe:iti:csd:2014:stored-function:provider-search

Response

Results are returned as a valid CSD document with a root document element of CSD. The results set is contained entirely within the providerDirectory element and consists of the full content
1310 of the provider elements of matching providers as maintained by the Care Services InfoManager.

Parameters

Query Parameters are defined (formatted as per the XFORMS core model specification) as the content of following elements relative to the requestParams child element of function:

1315 - id *csd:uniqueID*: (optional) If present and the @entityID attribute contains a non-empty value, it is an ID which uniquely identifies a provider. This is an exact match.

- `otherID` *csd:otherID*: (optional) If present and the `@assigningAuthorityName` attribute contains a non-empty value, then the result set is restricted to only those providers which have a `otherID` with the given `@assigningAuthorityName` and `@code`

1320
- `commonName` *xs:string*: (optional) If present and contains a non-empty value, then the result set shall be restricted to those providers which have a `demographic/name/commonName` this value. Case insensitive.

- `organizations` *element* (optional)

  - `organization` *element* (optional child element, multiple occurrences allowed) . Any
1325   `organization` child element of `organizations` has an `@entityID` attribute. If the set `organizations/organization/@entityID` in the request is non-empty, then the provider result set shall be further restricted to those providers such that there is a non-empty intersection between this set and the provider's attribute set `organizations/organization/@entityID`.

1330
- `facilities` *element* (optional)

  - `facility` *element* (optional child element, multiple occurrences allowed) . Any `facility` child element of `facilities` has an `@entityID` attribute. If the set `facilities/facility/@entityID` is non-empty, then the provider result set shall be further restricted to those providers such that there is a non-empty intersection
1335   between this set and the provider's attribute set `facilities/facility/@entityID`.

    - `service` *element* (optional child element, multiple occurrences allowed). Any `service` child element of `facility` has an `@entityID` attribute. If, for any of the `facilities/facility` elements, the set `./service/@entityID` is non-empty, then the provider result set shall further be restricted to those providers
1340   such that the provider's attribute set `./service/@entityID` has a non-empty intersection with the corresponding `facilities/facility` element.

- `codedType` *csd:codedtype*: If present and contains a non-empty value the result set shall be restricted to those providers whose `codeType/@code` equals this value for the coding scheme specified by the `@codingScheme` attribute. Case insensitive.

1345
- `address` *csd:address*: (optional) Contains of any-number of child `addressLine` elements as follows:

  - Text content *xs:string*: (optional) If present and contains a non-empty value, then the results set shall be restricted to those providers whose have an `addressLine` with specified `@component` containing this value exactly. Case insensitive.

1350
  - `@component` *xs:string* : (Required attribute) The component of the address we are searching. Case insensitive.

- `start` *xs:int*: (optional) The starting index for results returned. Defaults to 1, which indexes the first provider matching the search parameters

- max *xs:int*: (optional) The maximum number of results returned. A value of less than zero
1355 implies no maximum.

- record *csd:record* : (optional) A child element to limit results according to

- @status *xs:string*: (optional) If present and contains a non-empty value, the result set
shall be restricted to those providers whose record/@status equals this value. Case
insensitive.

1360 - @updated *xs:dateTime*: (optional) If present and contains a non-empty value, the
result set shall be restricted to those providers whose record/@updated is at least the
given value.

```
<careServicesRequest>
    <function urn='urn:ihe:iti:csd:2014:stored-function:provider-search'>
      <requestParams>
        <codedType code="2221" codingScheme="ISCO-08" />
        <address>
          <addressLine component='city'>Kigali</addressLine>
        </address>
        <max>5</max>
      </requestParams>
    </function>
</careServicesRequest>
```

1375 **Figure 3.73.4.1.2.2-1: Example Provider Search Request**

3.73.4.1.2.3 Facility Search

The careServicesFunction XML required to define the Facility Search careServicesFunction can
be found in ITI TF-2x: Appendix Y, Figure Y.3-3. For a non-normative XQuery implementation
1380 of the facility search, see ITI TF-2x: Appendix W.

Description

Performs a search for all facilities by name, type, address or ID.

The result set consists of all facilities matching the *search parameters* (id, primaryName, name,
codedType and address).

1385 The results set may be further restricted according to the *limit parameters* (start, max,
record/@status and record/@updated). An ordering of the result set is not specified.

Content-Type: text/xml
URN: urn:ihe:iti:csd:2014:stored-function:facility-search

Response

1390 Results are returned as a valid CSD document with a root document element of CSD. The results
set is contained entirely within the facilityDirectory element and consists of the full content
of the facility elements of matching facilities as maintained by the Care Services
InfoManager.

Parameters

1395   Query Parameters are defined (formatted as per the XFORMS core model specification) as the content of following elements relative to the `requestParams` child element of `function`:

- `id` *csd:uniqueID*: (optional) If present and the `@entityID` attribute contains a non-empty value, it is an ID which uniquely identifies a facility. This is an exact match.

- `otherID` *csd:otherID*: (optional) If present and the `@assigningAuthorityName` attribute contains a non-empty value, then the result set is restricted to only those organizations which have a `otherID` with the given `@assigningAuthorityName` and `@code`

- `primaryName` *xs:string*: (optional) If present and contains a non-empty value, then the result set shall be restricted to those facilities whose `primaryName` contains this value. Case insensitive.

- `name` *xs:string*: (optional) If present and contains a non-empty value, then the result set shall be restricted to those facilities which have a `primaryName` or `otherName` element containing this value. Case insensitive.

- `organizations` *element* (optional)

    - `organization` *element* (optional child element, multiple occurrences allowed) . Any `organization` child element of `organizations` has an `@entityID` attribute. If the set `organizations/organization/@entityID` in the request is non-empty, then the facility result set shall be further restricted to those facilities such that there is a non-empty intersection between this set and the facility's attribute set `organizations/organizations/@entityID`.

        - Any `service` child element of `organization` has an `@entityID` attribute. If, for any of the `organizations/organization` elements, the set`./service/@entityID` is non-empty, then the facility result set shall further be restricted to those facilities such that the facility's attribute set `./service/@entityID` has a non-empty intersection with the corresponding `organizations/organization` element.

- `codedType` *csd:codedtype*: If present and contains a non-empty value the result set shall be restricted to those facilities whose `codedType/@code` equals this value for the coding scheme specified by the `@codingScheme` attribute. Case insensitive.

- `address` *csd:address*: (optional) Contains of any-number of child `addressLine` elements as follows:

    - Text content *xs:string*: (optional) If present and contains a non-empty value, then the results set shall be restricted to those facilities whose have an `addressLine` with specified `@component` containing this value exactly. Case insensitive.

1430
- @component *xs:string* : (Required attribute) The component of the address we are searching. Case insensitive.

- start *xs:int*: (optional) The starting index for results returned. Defaults to 1, which indexes the first facility matching the search parameters

1435
- max *xs:int*: (optional) The maximum number of results returned. A value of less than zero implies no maximum.

- record *csd:record* : (optional) A child element to limit results according to

  - @status *xs:string*: (optional) If present and contains a non-empty value, the result set shall be restricted to those facilities whose record/@status equals this value. Case insensitive.

1440
  - @updated *xs:dateTime*: (optional) If present and contains a non-empty value, the result set shall be restricted to those facilities whose record/@updated is at least the given value.

1445

1450

```xml
<careServicesRequest>
    <function urn='urn:ihe:iti:csd:2014:stored-function:facility-search'>
      <requestParams>
       <codedType code='OPC-example' codingScheme="USDVA-example"/>
       <address>
          <addressLine component='city'>Chapel Hill</addressLine>
       </address>
       <max>5</max>
      <requestParams>
    </function>
</careServicesRequest>
```

1455
Figure 3.73.4.1.2.3-1: Example Facility Search

3.73.4.1.2.4 Service Search

The careServicesFunction XML required to define the Service Search careServicesFunction can be found in ITI TF-2x: Appendix Y, Figure Y.3-4. For a non-normative XQuery implementation
1460 of the service search, see ITI TF-2x: Appendix W.

Description

Performs a search for all services by type or ID.

The result set consists of all services matching the *search parameters* (id and codedType).

1465 The results set may be further restricted according to the *limit parameters* (start, max, record/@status and record/@updated). An ordering of the result set is not specified.

Content-Type: text/xml
URN: urn:ihe:iti:csd:2014:stored-function:service-search

Response

1470 Results are returned as a valid CSD document with a root document element of `CSD`. The results set is contained entirely within the `serviceDirectory` element and consists of the full content of the `service` elements of matching services as maintained by the Care Services InfoManager.

Parameters

Query Parameters are defined (formatted as per the XFORMS core model specification) as the
1475 content of following elements relative to the `requestParams` child element of `function`:

- `csd:id` *csd:uniqueID*: (optional) If present and the `@entityID` attribute contains a non-empty value, it is an ID which uniquely identifies a service. This is an exact match.

- `csd:otherID` *csd:otherID*: (optional) If present and the `@assigningAuthorityName` attribute contains a non-empty value, then the result set is restricted to only those
1480 organizations which have a `csd:otherID` with the given `@assigningAuthorityName` and `@code`

- `csd:codedType` *csd:codedtype*: If present and contains a non-empty value the result set shall be restricted to those services whose `codedType/@code` equals this value for the coding scheme specified by the `@codingScheme` attribute. Case insensitive.

1485 - `csd:start` *xs:int*: (optional) The starting index for results returned. Defaults to 1, which indexes the first service matching the search parameters

- `csd:max` *xs:int*: (optional) The maximum number of results returned. A value of less than zero implies no maximum.

- `csd:record` *csd:record* : (optional) A child element to limit results according to

1490 - `@status` *xs:string*: (optional) If present and contains a non-empty value, the result set shall be restricted to those services whose `record/@status` equals this value. Case insensitive.

- `@updated` *xs:dateTime*: (optional) If present and contains a non-empty value, the result set shall be restricted to those services whose `record/@updated` is at least the
1495 given value.

```
 <careServicesRequest>
    <function urn='urn:ihe:iti:csd:2014:stored-function:service'>
      <requestParams>
        <codedType code="76499" codingScheme="HCPCS" />
        <start>101</max>
      </requestParams>
    </function>
</careServicesRequest>
```

1500

1505 **Figure 3.73.4.1.2.4-1: Example Service Search**

3.73.4.1.3 Expected Actions

In response to receiving the submitted Find Matching Services Request, the Care Services InfoManager shall:

1510   1. Accept a parameterized query in a Find Matching Services message

2. Verify the required parameters are included in the request. Additionally, special rules documented in Table 3.73.4.1.2-2 shall be verified. The function shall be executed against the root node of the CSD document managed by the Care Services InfoManager. Any parameters passed to the function are accessible in the dynamic context of the
1515   function under the variable $careServicesRequest

3. Errors shall be returned for the conditions specified in Table 3.73.4.2.2-2 using the codes indicated in that table.

4. Process the query as appropriate and return a reply to the Service Finder containing the query result.

1520   For illustration, example requests are shown in Sections 3.73.4.1.2.1 to 3.73.4.1.2.4.

See ITI TF-2x: Appendix W for more examples of a Service Finder Query Request and Response.

3.73.4.1.4 Protocol Requirements

The transport protocol requirements described in this section apply to the request messages for
1525   both the Query for Updated Services [ITI-74] and Find Matching Services [ITI-73] transactions. These transactions are conveyed as defined by the appropriate `xforms:model/xforms:submission` element of the `csd:careServicesFunction` defining the Stored Query being invoked and according to the appropriate protocol as defined in Sections 3.73.4.1.4.1 through 3.73.4.1.4.3.

1530   ### 3.73.4.1.4.1 SOAP Protocol Requirements

The syntax of the SOAP transport is defined in the XFORMS `model` for the `csd:careServicesFunction` defintion of the stored function in ITI TF-2x: Appendix Y.3 by the elements:

- `xforms:instance[id='soap-request']`

1535   - `xforms:submission[id='soap-submission']`

- `xforms:bind[starts-with(@nodeset,"instance('soap-request')"]`

The requesting actor shall:

- put the query parameters for the desired Stored Query as a child element of `soap:Envelope/soap:Body/csd:careServicesRequest` relative to the
1540   `xforms:instance[id='soap-request']` element

- generate a UUID as a message ID as the text content of the
  `soap:Envelope/soap:Header/wsa:MessageID` relative to the
  `xforms:instance[id='soap-request']` element

1545
- submit the message, per the XFORMS core model specification, as defined by the
  `xforms:submission[id='soap-submission']`

The jurisdiction shall determine whether the http and/or https protocols may be used.

Implementors using the SOAP protocol shall comply with all requirements described in ITI TF-2x: Appendix V: Web Services for IHE Transactions. Transport via this protocol shall use SOAP 1.2. See ITI TF-2x: Appendix V for details. The following requirements apply to this
1550 transaction:

- The following types shall be imported (xs:import) in the `/definitions/types` section:

  - `namespace="urn:ihe:iti:csd:2013"`

  - `schema="CSD.xsd"` as provided in ITI TF-2x: Appendix Y.2

- The `definitions/message/part/@element` attribute of the request message shall be
1555   defined as "`csd:careServicesRequest`"

Refer to Table 3.73.4.1.4-1 for additional attribute requirements

Table 3.73.4.1.4-1: Additional Attribute Requirements

Attribute	Value
/definitions/portType/operation/input/@wsaw:Action	`urn:ihe:iti:csd:2013:GetCareServicesRequest`
/definitions/portType/operation/output/@wsaw:Action	`urn:ihe:iti:csd:2013:GetCareServicesResponse`
/definitions/binding/operation/wsoap12:operation/@soapActionRequired	false

1560 These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Sample request and response messages for the transactions are shown in Figures 3.74.4.1.2-1 and 3.74.4.2.2-1 respectively.

For an informative WSDL for the responding actor (i.e., the Care Services InfoManager for [ITI-
1565 73] or the Care Serivces Directory for [ITI-74]), see ITI TF-2x: Appendix W.

The `csd:careServicesRequest` element is defined Figure 3.73.4.1.2-2.

A full XML Schema Document for the CSD types is available online on the IHE FTP site, see ITI TF-2x: Appendix W.

3.73.4.1.4.2 HTTP POST Protocol Requirements

1570 The syntax of the HTTP POST transport is defined in the XFORMS `model` for the `csd:careServicesFunction` defintion of the stored function in ITI TF-2x: Appendix Y3 by the elements:

- `xforms:instance[id='http-post-request']`

- `xforms:submission[id='http-post-submission']`

1575 - `xforms:bind[starts-with(@nodeset,"instance('http-post-request')"]`

The requesting actor shall:

- put the query parameters for the desired Stored Query as a child element of the

  `xforms:instance[id='http-post-request']` element

- submit the message, per the XFORMS core model specification, as defined by the
1580 `xforms:submission[id='http-post-submission']`

The jurisdiction will determine whether the http and/or https protocols may be used.

3.73.4.1.4.3 HTTP GET Protocol Requirements

The syntax of the HTTP GET transport is defined in the XFORMS `model` for the `csd:careServicesFunction` defintion of the stored function in Appendix Y.3 by the elements:

1585 - `xforms:instance[id='http-get-request']`

- `xforms:submission[id='http-get-submission']`

- `xforms:bind[starts-with(@nodeset,"instance('http-get-request')"]`

The requesting actor shall

- put the query parameters for the desired Stored Query as a child element of the

1590 `xforms:instance[id='http-get-request']` element

- submit the message, per the XFORMS core model specification, as defined by the
`xforms:submission[id='http-get-submission']`

The jurisdiction will determine whether the http and/or https protocols may be used.

3.73.4.2 Find Matching Services Response (Stored Query)

1595 **3.73.4.2.1 Trigger Events**

The Care Services InfoManager prepares and returns a Find Matching Services response upon receiving an inbound Find Matching Services query.

3.73.4.2.2 Message Semantics

The Care Services InfoManager shall return a Find Matching Services response in the HTTP
1600 transaction reply.

For all stored queries described in Sections 3.73.4.1.2.1 to 3.73.4.1.2.4, the query response shall be an XML document containing the `CSD` element as defined by the CSD XML Schema (see ITI TF-2x: Appendix Y2).

The message semantics for a Find Matching Services [ITI-73] response shall be a Stored Query as defined Section 3.73.4.1.1. The response shall be conveyed using the HTTP POST or SOAP transport protocols as defined in Section 3.73.4.2.4 subject to the following:

- A Care Services InfoManager receiving a Find Matching Services [ITI-73] request shall support all of transport protocols defined by the `xforms:submission` elements of the `csd:careServicesFunction` for the Stored Query it is receiving. The response message shall be conveyed using the same transport protocol (ie HTTP or SOAP) as was used by the Services Finder initiating the transaction.

The response content shall be returned according to the requirements of the transport protocol as defined in Section 3.73.4.2.4.

Potential error conditions are defined in Table 3.73.4.2.4-1 and shall be conveyed according to requirements of the transport protocol.

The elements in the resulting XML content are defined in the CSD XML Schema Definition found in ITI TF-2x: ITI TF-2x: Appendix Y.2 and further elaborated upon by the normative content tables in ITI TF-2x: Appendix Y.1.

3.73.4.2.3 Expected Actions

There is no defined expected action to be taken by the Service Finder once the response has been received. The Service Finder processes the response in accordance with the functions of its application.

3.73.4.2.4 Protocol Requirements

The transport protocol requirements described in this section apply for the response messages for both the Query for Updated Services [ITI-74] and Find Matching Services [ITI-73] transactions. These transactions are conveyed as defined by the appropriate `xforms:model/xforms:submission` element of the `csd:careServicesFunction` defining the Stored Query being invoked.

Upon excuting a Stored Function, the responding actor shall prepare the `xforms:instance[id='care-services-result']` response element as follows:

- The appropriate attribute response code shall be put in the `csd:careServicesResponse/@code` attribute. The response codes in Table 3.73.4.2.4-1 are used for all the transport protocols as specified in Sections 3.73.4.2.4.1 – 3.73.4.2.4.3.

- A transaction response UUID shall be generated and placed in the `csd:careServicesResponse/@id` attribute conformant to RFC 4122.

- If the query was executed successfully, the query results shall be placed as child elements of the `csd:careServicesResponse/csd:result` element.

- The value of the `csd:careServicesRequest/@content-type` attribute shall be set to be the same as the `csd:careServicesFunction/@content-type` attribute, unless otherwise required by the Stored Query. See, for example, Section 3.73.4.3.2.1.

1640

- The responding actor shall dispatch an `xforms-submit-done` event to be processed according to the rules defined for the relevant transport protocol as specified in Sections 3.73.4.2.4.1 – 3.73.4.2.4.3.

1645

Table 3.73.4.2.4-1 Transaction Response Codes

Response Code	Trigger	Result
200	successful execution of query	The raw result of executing the query that was indicated in the requested Stored Query is returned
422	error executing query	An optional error message (implementation defined) is. The error message is implementation specific
415	an XML document was not received	An optional error message (implementation defined) is. The error message is implementation specific
400	the received XML document is not syntactically correct	An optional error message (implementation defined) is. The error message is implementation specific
404	no `csd:careServicesFunction` with the specified URN is registered.	An optional error message (implementation defined) is. The error message is implementation specific

3.73.4.2.4.1 SOAP Protocol Requirements

The syntax of the SOAP transport is defined in the XFORMS `model` for the `csd:careServicesFunction` defintion of the stored function in Appendix Y.3. Once the

1650 `xforms:instance[id='care-services-result']` response has been prepared, the responding actor shall:

- process the actions for the `xforms-submit-done` event as defined under the `xforms:actions[id='soap-prepare']` element

- return the resulting content of the `xforms:instance[id='soap-response']` to the
1655 requesting actor

Implementations shall comply with all requirements described in ITI TF-2x: Appendix V: Web Services for IHE Transactions. Transport via this protocol shall use SOAP1.2.

3.73.4.2.4.2 HTTP POST Protocol Requirements

The syntax of the HTTP POST transport is defined in the XFORMS `model` for the
1660 `csd:careServicesFunction` defintion of the stored function in ITI TF-2x: Appendix Y.3. Once the response `xforms:instance[id='care-services-result']` has been prepared, the responding actor shall:

- process the actions for the `xforms-submit-done` event as defined under the `xforms:actions[id='http-post-prepare']` element

1665 - return the resulting content of the `xforms:instance[id='http-post-response']` to the requesting actor as defined by the relative `http:result` element according to the EXPath HTTP Client Module.

3.73.4.2.4.3 HTTP GET Protocol Requirements

The syntax of the HTTP GET transport is defined in the XFORMS `model` for the
1670 `csd:careServicesFunction` defintion of the stored function in ITI TF-2x: Appendix Y.3. Once the response `xforms:instance[id='care-services-result']` has been prepared, the responding actor shall:

- Process the actions for the `xforms-submit-done` event as defined under the `xforms:actions[id='http-get-prepare']` element

1675 - return the resulting content of the `xforms:instance[id='http-get-response']` to the requesting actor as defined by the relative `http:result` element according to the EXPath HTTP Client Module.

3.73.4.3 Find Matching Services Request (Ad hoc XQuery Option)

A Service Finder or Care Services InfoManager that supports the Ad hoc XQuery Option shall
1680 support the Ad hoc XQueries described in Sections 3.73.4.3 and 3.73.4.4.

In the definition of the Find Matching Services Request the following XML namespaces are used.

Table 3.73.4.3-1: Namespace Definitions

csd	urn:ihe:iti:csd:2013

1685 The `csd` namespace is also as used as the default element namespace for describing elements of "CSD" and "Find Matching Services Request" documents

3.73.4.3.1 Trigger Event

A Service Finder Actor sends a Find Services Request (Ad hoc XQuery) when it needs to access service information maintained by the Care Services InfoManager and where there is not already a suitable stored query for this purpose.

3.73.4.3.2 Message Semantics

The message semantics for a Find Matching Services [ITI-73] request shall be a Stored Query as defined Section 3.73.4.1.1. The Service Finder shall convey the Find Matching Services request using either the HTTP POST or SOAP transport protocols as defined in Section 3.73.4.1.4

The semantic definition of the query parameter for the Stored Query is defined in Section 3.73.4.3.2.1.

The `csd:careServicesFunction` XML required to define the Adhoc Option can be found in ITI TF-2x: Appendix Y, Figure Y.3-1 – Y.3-4. For a non-normative XQuery implementation, see ITI TF-2x: Appendix W.

3.73.4.3.2.1 Query Parameters

Query Parameters are defined (formatted as per the XFORMS core model specification) as the content of the `csd:adhoc` element.

The `csd:dhoc` element has two mandatory properties:

1. `text()`: The text content of the adhoc node shall be a well-formed XQuery conformant to the standards referenced in Section 3.73.3.

2. `@content-type`: This attribute is employed by the Service Finder to indicate the mime-type of the result of execution of the XQuery. Values must be conformant to:

   - http://tools.ietf.org/html/rfc2045

   - http://tools.ietf.org/html/rfc2046

An example Ad hoc XQuery is shown below. Note, this example XQuery returns content as may be requested per the US Office of the National Coordinator's Provider query #7 (http://wiki.siframework.org/file/view/ESI%20Query%20and%20Response.pdf).

```
 <csd:careServicesRequest>
  <csd:function urn="urn:ihe:iti:csd:2014:adhoc" >
   <csd:adhoc content-type="text/xml"><![CDATA[
    (:
       This is an example query which searches for a Genetic Counselor (according to the BCBSIL
       coding scheme) with a last name starting with "L" and which is part of a Hospital
       organizational network  (according to the NCDHHS coding scheme) which is based in the
       27599 zip code.

       The query will be executed against the root element of the CSD document.
       It is designed as a template where specific search parameters can be put into the
       $orgSearch and $provSearch

    :)
    declare default element  namespace   "urn:ihe:iti:csd:2013";
    import module namespace csd  "urn:ihe:iti:csd:2013" at "csd_base_library.xq";
    let $orgSearch := document{
      <requestParams>
       <id/>
       <primaryName/>
       <name>University</name>
       <codedType cod"="HL-examp"e" codingSchem"="ncdhhs.gov-examp"e"/>
       <address>
         <addressLine component="zipcode">27599</addressLine>
       </address>
      </requestParams>
     }
    let $provSearch := document{
      <requestParams>
       <id/>
       <otherId/>
       <forename/>
       <surname>L</surname>
       <codedType code"="DA-example" codingScheme="BCBSIL-example"/>
       <address/>
      </requestParams>
     }
    <CSD xmlns:csd="urn:ihe:iti:csd:2013">{
      let $orgs0 := csd:filter_by_primary_id(/CSD/organizationDirectory/*,$orgSearch/params/id)
      let $orgs1 := csd:filter_by_primary_name($orgs0,$orgSearch/params/primaryName)
      let $orgs2 := csd:filter_by_name($orgs1,$orgSearch/params/name)
      let $orgs3 := csd:filter_by_coded_type($orgs2,$orgSearch/params/codedType)
      let $orgs4 := csd:filter_by_address($orgs3, $orgSearch/address/addressLine)
      let $orgs5:= csd:join_parent_organizations($orgs4,/CSD/organizationDirectory)
                       union
                       csd:join_child_organizations($orgs4,/CSD/organizationDirectory)
      let $provs0 := csd:filter_by_primary_id(/CSD/providerDirectory/*,$provSearch/params/id)
      let $provs1 := csd:filter_by_other_id($provs0,$provSearch/params/otherID)
      let $provs2 := csd:filter_by_forename($provs1,$provSearch/params/forename)
      let $provs3 := csd:filter_by_surname_starts_with($provs2,$provSearch/params/surname)
      let $provs4 := csd:filter_by_coded_type($provs3,$provSearch/params/codedType)
      let $provs5 := csd:filter_by_address($provs4, $provSearch/params/address/addressLine)
      let $provs6:= csd:filter_by_organizations($provs5,$orgs5)
      return  (
      <organizationDirectory>{$orgs5}</organizationDirectory> ,
      <serviceDirectory/> ,
      <facilityDirectory/> ,
      <providerDirectory>{$provs6}</providerDirectory>  )
      }</CSD>
   </csd:adhoc>
  </csd:function>
</csd:careServicesRequest>
```

Figure 3.73.4.3.2-1: Example Ad Hoc Query

The calling convention for submitting an Ad hoc XQuery using the Find Matching Service Request message is the same as in Table 3.73.4.1.2-2.

3.73.4.3.3 Expected Actions

In response to receiving the submitted Find Matching Services Request, the Care Services InfoManager shall:

1. Accept an Ad hoc query in a Find Matching Services message. If the Care Services InfoManager does not support the Ad hoc XQuery Option it shall immediately return an error 422 (see Table 3.73.4.2.2-2) with a message indicating the Ad hoc XQuery Option is not supported.

2. "Sanitize" the inbound XQuery query using the techniques described in the **Specific Security Considerations Regarding the Use of XQuery** subsection of ITI TF-1:35.5.

3. Process the query as appropriate and return a result to the Service Finder containing the query results or error results as described in Table 3.73.4.2.2-2.

3.73.4.4 Find Matching Services Response (Ad hoc XQuery Option)

A Service Finder or Care Services InfoManager that supports the Ad hoc XQuery Option shall support the Ad hoc XQueries described in Sections 3.73.4.3 and 3.73.4.4.

3.73.4.4.1 Trigger Event

The Care Services InfoManager prepares and returns a Find Matching Services response upon its receipt of an inbound Find Matching Services query.

3.73.4.4.2 Message Semantics

The Care Services InfoManager shall respond to a Find Matching Services request according to Section 3.73.4.2.2. In addition, the mime-type returned by the Care Services InfoManager for an unencapsulated request should be the same as the value of the `csd:adhoc/@content-type` attribute sent in the request.

3.73.4.4.3 Expected Actions

There is no defined expected action to be taken by the Service Finder once the response has been received. The Service Finder processes the response in accordance with the functions of its application.

3.73.5 Security Considerations

For Care Services InfoManager actors that support the Ad hoc XQuery Option, care shall be taken to appropriately guard against inject attacks by employing the techniques described in the **Specific Security Considerations Regarding the Use of XQuery** subsection of ITI TF-1:35.5.

1810 3.73.5.1 Security Audit Considerations

To be determined by CSD Profile implementers. NOTE: Because this transaction does not convey content containing protected health information (PHI) there are no stipulated audit requirements.

1815

3.74 Query for Updated Services [ITI-74]

3.74.1 Scope

The Query for Updated Services transaction is used to obtain the directory records that have been updated since the refresh timestamp provided in the query.

1820 **3.74.2 Use Case Roles**
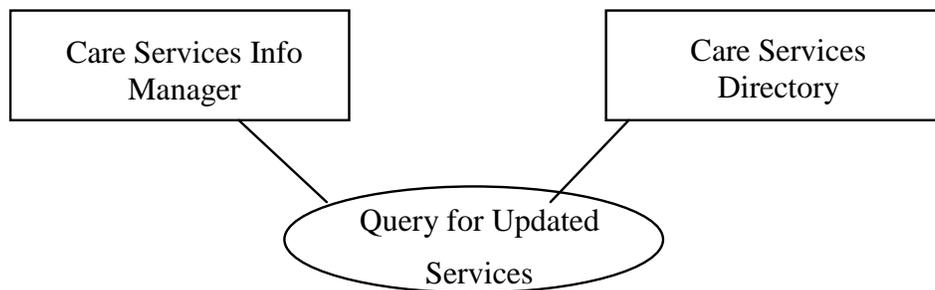
Figure 3.74.2-1: Basic Process Flow for Query for Updated Services

Table 3.74.2-1: Actor Roles

Actor:	Care Services InfoManager
Role:	On a periodic basis to be determined by the implementing jurisdiction, the Care Services InfoManager shall submit a Query for Updated Services [ITI-74] to one or more Care Services Directory actors.
	The query has a single timestamp parameter indicating the time of the last refresh. The XML document returned by the Directory Actor(s) shall be used to update the InfoManager's cached content.
Actor:	Care Services Directory
Role:	The Care Services Directory receives from the Care Services InfoManager a Query for Updated Services [ITI-74] indicating the timestamp of the last directory content refresh. The Care Services Directory Actor shall respond to the inbound transaction and return an XML document consistent with the CSD XML schema (See ITI TF-2x: Appendix W). This XML document will contain updates and inserts applied to the Directory's content since the last refresh.

1825

In response to the received Query for Updated Services message, the Care Services Directory Actor prepares and returns an XML document containing all directory content which has been inserted or updated since the specified refresh timestamp. The format of this XML document shall conform to the CSD XML schema (see ITI TF-2x: Appendix W).

1830 **3.74.3 Referenced Standards**

This transaction makes use of the standards listed in Section 3.73.3.

3.74.4 Interaction Diagram
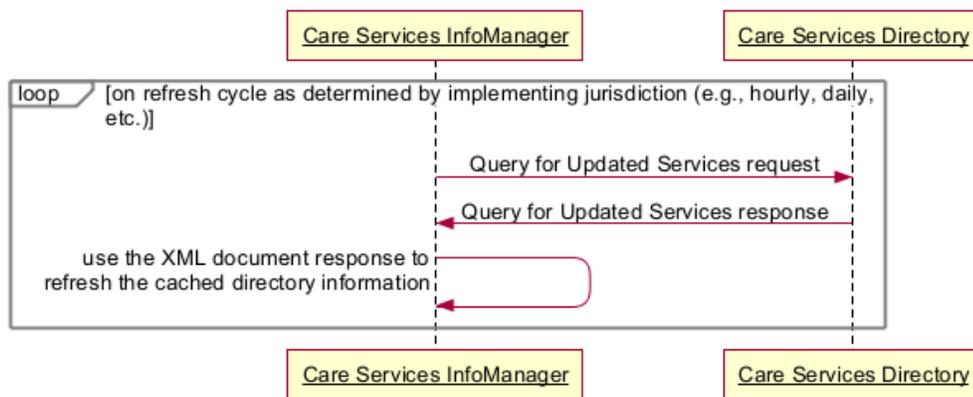

```
Diagram pseudocode:

participant "Care Services InfoManager"
participant "Care Services Directory"

loop on refresh cycle as determined by implementing jurisdiction (e.g., hourly, daily, etc.)
 Care Services InfoManager -> Care Services Directory: Query for Updated Services request Care
Services Directory -> Care Services InfoManager: Query for Updated Services response  Care
Services InfoManager -> Care Services InfoManager: use the XML document response to \nrefresh the
cached directory information
end
```

1835

1840

1845 **Figure 3.74.4-1: Query for Updated Services [ITI-74] Interactions**

3.74.4.1 Query for Updated Services Request

3.74.4.1.1 Trigger Event

The Care Services InfoManager sends a Query for Updated Services request message to the one
1850 or more Care Services Directories in its implementation-specific "network" on a periodic basis. The polling period and the list of participating directories are determined by the implementation and shall be configurable on the Care Services InfoManager.

3.74.4.1.2 Message Semantics

The message semantics for a Find Matching Services [ITI-73] request shall be a Stored Query as
defined in Section 3.73.4.1.1. The request shall be conveyed using the HTTP GET or SOAP
transport protocols as defined in Section 3.73.4.1.4 subject to the following:

* A Care Services InfoManager initiating a Find Matching Services [ITI-73] request shall
  support at least one of transport protocols defined by the `xforms:submission` element
  of the `csd:careServicesFunction` for the Query For Updated Services Stored Query it
  is invoking.

The semantic definitions of the query parameter for the Stored Query is defined in Section
3.73.4.3.2.1.

The `csd:careServicesFunction` XML required to define the Find Matching Services Stored
Query can be found in ITI TF-2x: Appendix Y, Figure Y.3-6. For a non-normative XQuery
implementation, see ITI TF-2x: Appendix W.

3.74.4.1.2.1 Query Parameters

Query Parameters are defined (formatted as per the XFORMS core model specification) as the
content of following elements relative to the `csd:requestParams` element:

* `csd:updated` *xs:dateTime*: Contains a non-empty value, the result set should be
  restricted to those entities whose `csd:record/@updated` is at least the given value.

An example message is shown in Figure 3.74.4.1.2.1-1.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:csd="urn:ihe:iti:csd:2013">
    <soap:Header>
        <wsa:Action soap:mustUnderstand="1"
            >urn:ihe:iti:csd:2013:GetCareServicesRequest</wsa:Action>
        <wsa:MessageID>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</wsa:MessageID>
        <wsa:ReplyTo soap:mustUnderstand="1">
            <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To soap:mustUnderstand="1">http://host/csDirectory</wsa:To>
    </soap:Header>
    <soap:Body>
        <csd:careServicesRequest
            urn='urn:ihe:iti:csd:2016:stored-function:query-for-updated-services'>
          <csd:requestParams>
              <csd:lastModified>2002-05-30T09:30:10.5</csd:lastModified>
          <csd:requestParams>
        </csd:careServicesRequest>
    </soap:Body>
</soap:Envelope>
```

Figure 3.74.4.1.2.1-1: Example Query for Updated Services SOAP Request Message

To initiate the Query for Updated Services request, the Care Services InfoManager shall submit
the message containing the timestamp of its last refresh to the endpoint exposed by each Care

---

Services Directory Actor in the set of Care Services Directories that the Care Services InfoManager polls. How the Care Services InfoManager of manages this set of Care Services Directory endpoints is implementation specific and not defined by IHE.

3.74.4.1.3 Expected Actions

Upon receipt of an inbound Query for Updated Services message, the Care Services Directory shall:

1. Determine which of its directory records have been updated or inserted since the timestamp passed in the inbound message. A "directory record" is defined as an organization, service, facility, or provider element as defined in the CSD XML Schema Definition. There may be zero or more records that have been altered or inserted since the timestamp.

2. Where a new directory record has been inserted, indicate the insertion timestamp in the `record/@created` element returned in a Query for Updated Services response

3. Where a directory record has been altered, indicate the update timestamp in the `record/@updated` element returned in a Query for Updated Services response

4. Place implementation-specific extensions to the CSD XML schema under the appropriate `//extension` element.

5. Express the inserted and updated content as an XML document consistent with the CSD XML Schema (see ITI TF-2x: Appendix W) and return this XML file to the Care Services InfoManager in a Query for Updated Services response message.

3.74.4.1.4 Protocol Requirements

Implementors of this transaction shall comply with all requirements described in Section 3.74.3.1.4 for SOAP and HTTP GET.

3.74.4.2 Query for Updated Services Response

3.74.4.2.1 Trigger Events

The Care Services Directory Actor shall respond to an inbound Query for Updated Services request message when it receives one from a Care Services InfoManager.

3.74.4.2.2 Message Semantics

The message semantics for a Query for Updated Services request shall be a Stored Query as defined Section 3.73.4.1.1. If the Care Services Directory actor supports the transport protocol of the request, then the reply message shall be conveyed using the same transport protocols as was used by the Care Services InfoManager actor initiating the transaction.

The content returned by the Care Services Directory will be a `csd:CSD` element conformant with the CSD XML Schema (see ITI TF-2x: Appendix Y.2 and ITI TF-2x: Appendix W). The

response content shall be returned according to the requirements of the transport protocol as defined in Section 3.73.4.2.4.

An example Query for Updated Services response message is shown in Figure 3.74.4.2.2-1

1935

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:csd="urn:ihe:iti:csd:2013">
    <soap:Header>
        <wsa:Action soap:mustUnderstand="1"
            >urn:ihe:iti:csd:2013:GetCareServicesResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:060bf1f5-a5d8-4741-86c6-cb240f887e5b</wsa:MessageID>
        <wsa:To soap:mustUnderstand="1"> http://www.w3.org/2005/08/addressing/anonymous</wsa:To>
        <wsa:RelatesTo>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</wsa:RelatesTo>
    </soap:Header>
    <soap:Body>
        <csd:careServicesResponse code="200">
          <csd:result>
            <csd:CSD >
                <csd:organizationDirectory/>
                <csd:serviceDirectory/>
                <csd:facilityDirectory/>
                <csd:providerDirectory>
                    <csd:provider urn="urn:uuid: 79200711-d254-45fb-868e-060e40906ddf">
                        <!--Details of provider record go here-->
                    </csd:provider>
                </csd:providerDirectory>
            </csd:CSD>
          </csd:result>
        </csd:careServicesResponse>
    </soap:Body>
</soap:Envelope>
```

1940

1945

1950

1955

1960

Figure 3.74.4.2.2-1: Example Query for Updated Services SOAP Response Message

1965 ### 3.74.4.2.3 Expected Actions

Upon receipt of a Care Services Directory Actor's query response, a Care Services InfoManager shall:

1. Confirm the inbound XML document conforms to the CSD XSD.

2. If the query response is non-conformant to the CSD XML Schema (see ITI TF-2x: 1970   Appendix W) then it shall be ignored. See Section 3.74.4.2.4.

3. If the query response is conformant to the CSD XML schema, the Care Services InfoManager shall update its cached content to reflect the received result.

- Any directory record conveyed in the query response shall replace and overwrite in the cached content of an existing directory the record previously conveyed with an 1975   identical @urn attribute. By definition, a directory record shall be any elements matching the following xpath expressions:

  - /CSD/oraganizationDirectory/organization

  - /CSD/facilityDirectory/facility

- `/CSD/providerDirectory/provider`

1980
- `/CSD/serviceDirectory/service`

- For all cache inserts or updates, the Care Services InfoManager shall set the `record/@sourceDirectory` attribute to the URI of the Care Services Directory that was the source of the content.

4. Ensure that all cached directory records have unique IDs and unbroken cross-references.
1985   See Section 3.74.4.2.4.

3.74.4.2.4 Error Handling

This section describes the error conditions that shall be handled by the Care Services InfoManager.

3.74.4.2.4.1 Non-conformant responses

1990 A Care Services InfoManager's shall no longer poll any Care Services Directory that submits a non-conformant XML document response and shall stop sending Query for Updated Services Request messages to that Care Services Directory until an intervention has been made to address the issue. IHE does not define the remedial actions to address this fault condition.

3.74.4.2.4.2 Duplicate Unique IDs

1995 A Care Services InfoManager shall ensure that all directory records in its content cache have unique IDs. This means that no two Care Services Directory actors may submit a directory record with the same oid attribute. A Care Services InfoManager that contains cached directory records with duplicate IDs shall remain offline until this situation has been remediated. Care Services InfoManagers in an offline state shall respond to all inbound Find Matching Services
2000 request messages with an error message 422 plus appropriate message text indicating its fault condition (duplicate IDs). IHE does not define the remedial actions to address this fault condition.

3.74.4.2.4.3 Broken ID cross references

A Care Services InfoManager shall ensure that there are no broken ID cross references in its
2005 content cache. Cross-reference consistency means that:

- A matching `/CSD/serviceDirectory/service/@entityID` element shall exist for every `//service/@entityID` element in the cache.

- A matching `/CSD/facilityDirectory/facility/@entityID` element shall exist for every `//facility/@entityID` element in the cache.

2010
- A matching `/CSD/organizationDirectory/organization/@entityID` element shall exist for every `//organization/@entityID` element in the cache.

- A matching `/CSD/providerDirectory/provider/@entityID` element shall exist for every `//provider/@entityID` element in the cache.

2015   A Care Services InfoManager that contains cached directory records with broken ID cross-references shall remain offline until this situation has been remediated. A Care Services InfoManager in an offline state shall respond to all inbound Find Matching Services request messages with an error message 422 plus appropriate message text indicating its fault condition (broken ID cross references). IHE does not define the remedial actions to address this fault condition.

2020   **3.74.5 Security Considerations**

This transaction may be subject to the ATNA requirements for node authentication, access control, and encryption depending upon the endpoint actors. If the Care Services Directory and Care Services InfoManager are grouped with either an ATNA Secure Node or Secure Application, then the transaction shall be in accordance with the ATNA requirements.

2025   **3.74.5.1 Security Audit Considerations**

None mandated by this transaction.

3.75 Query for FreeBusy [ITI-75]

2030 ### 3.75.1 Scope

The Query for FreeBusy [ITI-75] transaction is used by a Service Finder that supports the FreeBusy Option to express a well-formed FreeBusy REPORT query to a CalDAV conformant Service Availability Actor adherent to the IETF RFC 4791 specification.

3.75.2 Use Case Roles
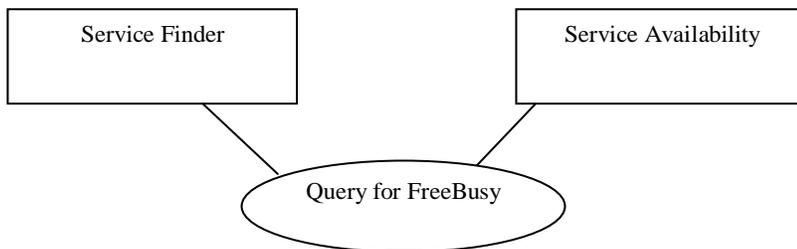
2035

Figure 3.75.2-1: Basic Process Flow for Query for FreeBusy

Table 3.75.2-1: Actor Roles

Actor:	Service Finder
Role:	A Service Finder Actor that supports the FreeBusy Option submits a CalDAV FreeBusy REPORT request to obtain FreeBusy information from a Service Availability Actor. The BUSY time is returned by the Service Availability Actor in the format stipulated in the RFC 4791 specification.
Actor:	Service Availability
Role:	The Service Availability Actor receives from the Service Finder a well-formed CalDAV FreeBusy REPORT request conformant to the IETF RFC 4791 specification. The Service Availability Actor processes the inbound query and returns the result in the format stipulated in the RFC 4791 specification.

3.75.3 Referenced Standards

RFC 4791   Calendaring Extensions to WebDAV (CalDAV)
                (http://tools.ietf.org/html/rfc4791).

3.75.4 Interaction Diagram
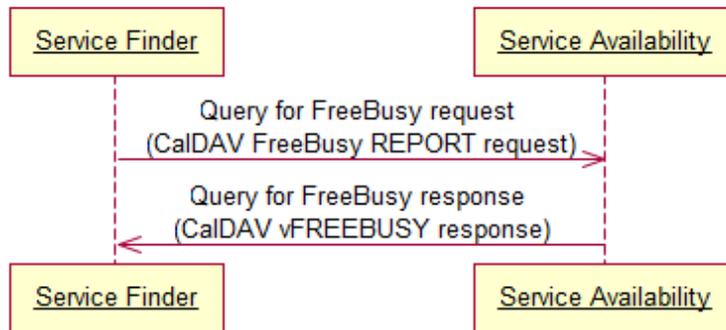

```
Diagram pseudocode:

participant "Service Finder"
participant "Service Availability"

Service Finder -> Service Availability: Query for FreeBusy request \n(CalDAV FreeBusy REPORT
request)
Service Availability -> Service Finder: Query for FreeBusy response \n(CalDAV vFREEBUSY response)
```

Figure 3.75.4-1: Query for FreeBusy Interactions

3.75.4.1 Query for FreeBusy Request

A Service Finder Actor that supports the FreeBusy Option submits a CalDAV FreeBusy REPORT request to obtain FreeBusy information from a Service Availability Actor.

3.75.4.1.1 Trigger Events.

A Service Finder will submit a Query for FreeBusy request to a Service Availability Actor when it wishes to obtain BUSY information about a Provider or Service at a particular Facility.

3.75.4.1.2 Message Semantics

The Query for FreeBusy request shall be constructed as a well-formed CalDAV FreeBusy REPORT request as specified in RFC 4791.

FreeBusy queries return "busy" data; therefore, the operating hours will be required in order to calculate the "free" time during an interval of interest. A Service Finder, in order to optionally

calculate "free" time, will either have to know ahead of time or obtain via the [ITI-73] transaction, the endpoint address for the FreeBusy service and the operating hours of the service.

In the case of an organization-offered service at a facility, this content is found here:

2070
- The `/CSD/facilityDirectory/facility/organizations/organization/service` element contains the operating hours and FreeBusy address for a specific service offered under the auspices of specific organization at a specific facility

In the case of a specific provider-offered service, the content is found here:

- The `/CSD/providerDirectory/provider/facilities/facility/service` element contains the operating hours and FreeBusy address for specific provider's services
2075
  offered at a specific facility.

When a Service Finder determines schedulable time for a health worker providing a service at a facility, the following precedence rules apply:

- If any `/CSD/providerDirectory/provider/facilties/facility/service/operatingHou rs` exist, they take precedence over any of the other elements defined below. The other
2080
  elements below are ignored.

- If any `/CSD/providerDirectory/provider/facilties/facility/operatingHours` exist, then they take next precedence over any of the other elements defined below. The other elements below are ignored.

2085
- If any `/CSD/facilityDirectory/facility/operatingHours` exists, then they are used for determine the schedulable time.

- If none of the above elements exist, then the Service Finders assumes that there are no restrictions on schedulable time.

2090
The Service Finder shall specify the CalDAV calendar address (as represented in the `freeBusyURI` element) and the time interval of interest and send a FreeBusy REPORT request in accordance with the IETF RFC 4791 specification.

An example FreeBusy query is shown below. In this example, the CalDAV endpoint is indicated as "example.com/drblack/work/" and the date range is indicated as "20060104T140000Z" to
2095
"20060105T220000Z".

```
REPORT example.com/drblack/work/ HTTP/1.1
Content-Type: application/xml; charset="utf-8"

<?xml version="1.0" encoding="utf-8" ?>
<C:free-busy-query xmlns:C="urn:ietf:params:xml:ns:caldav">
    <C:time-range start="20060104T140000Z"
        end="20060105T220000Z"/>
</C:free-busy-query>
```

2100

2105

3.75.4.1.3 Expected Actions

A Service Availability Actor will respond with a CalDAV FreeBusy "report" conformant to
IETF RFC 4791. If the request was not compliant with RFC 4791, the response will be an error
report.

2110

3.75.4.2 Query for FreeBusy Response

The Service Availability Actor processes the inbound query and returns the result.

3.75.4.2.1 Trigger Events

A Service Availability Actor will process an inbound Query for FreeBusy request when it
receives one from a Service Finder Actor.

2115

3.75.4.2.2 Message Semantics

A Query for FreeBusy response shall adhere to the format for FreeBusy responses defined in
IETF RFC 4791. An example of such a response is shown below. If it does not, the Service
Availability Actor shall respond with an error in accordance with RFC 4791.

2120

```
HTTP/1.1 200 OK
Date: Sat, 11 Nov 2006 09:32:12 GMT
Content-Type: text/calendar

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Healthcare Corp.//CalDAV Server//EN
BEGIN:VFREEBUSY
DTSTAMP:20050125T090000Z
DTSTART:20060104T140000Z
DTEND:20060105T220000Z
FREEBUSY;FBTYPE=BUSY-TENTATIVE:20060104T150000Z/PT1H
FREEBUSY:20060104T190000Z/PT1H
END:VFREEBUSY
END:VCALENDAR
```

2125

2130

2135

3.75.4.2.3 Expected Actions

There is no defined expected action to be taken by the Service Finder once the response has been
received. The Service Finder processes the response in accordance with the functions of its
application.

2140 ### 3.75.5 Error Handling

Errors will be returned by the Service Availability Actor to the Service Finder for the conditions and in the manner described in IETF RFC 4791.

3.75.6 Security Considerations

This transaction is subject to the security and privacy constraints described in Sections 6.1.1 and
2145 7.10 of the RFC 4791 specification (http://tools.ietf.org/html/rfc4791#section-6.1.1).

3.75.6.1 Security Audit Considerations

There are no specific audit requirements for this transaction. The schedule information made available from this service is not generally considered private information.

2150

Volume 2x – Appendices

Add Appendix Y

Appendix Y – Care Service Discovery (CSD) Data Model (Normative)

Y.1   Care Service Discovery – Entities and Attributes

2155   The following diagrams and related tables describe the entities and attributes that are discoverable using the Care Services Discovery Profile. This profile deals with four "directory" entities plus an optional FreeBusy entity:

1. Organization – Organizations are "umbrella" entities; these may be considered the administrative bodies under whose auspices care services are provided such as hospitals, clinics, Healthcare Information Exchanges (HIEs), Integrated Delivery Networks (IDNs), non-government Organizations (NGOs), faith-based organizations (FBOs) or a physician's family practice. An organization has a unique identifier and administrative attributes such as contact person, mailing address, etc. An organization may be related to a parent organization (Table Y.1-1).

2160

2165

2. Facility – Facilities are physical care delivery sites such as hospitals, clinics, health outposts, physician offices, labs, pharmacies, etc. A Facility has a unique identifier, geographic attributes (address, geocode), contact attributes, attributes regarding its hours of operation, etc. Each Facility is related to at least one Organization and to at least one Service. A Facility may be related to one or more Providers (Table Y.1-2).

2170

3. Service – A Service identifies a particular care service; each service has a unique identifier. Examples include: surgical services, antenatal care services or primary care services. The combination of a Service offered at a Facility (FacilityService) may have specific attributes including language spoken or understood at that facility, hours of operation, etc. (Table Y.1-3).

2175

4. Provider – A Provider is a health worker (see WHO definition), such as a physician, nurse, pharmacist or district health manager. Providers have contact and demographic attributes, etc. Each Provider is related to at least one Organization, Facility or FacilityService. Specific attributes may be associated with the Provider relationship with any of these three entities (such as hours of operation, FreeBusy, etc., Table Y.1-4).

2180

5. FreeBusy – The optional FreeBusy entity returns the "busy" information for a FacilityService or for a Provider. It is up to the calling application (Service Finder) to use the appropriate entity's hours of operation to calculate schedule availability from this "busy" information (and such calculations are beyond the scope of this profile).

<u>Organization</u>

2185   The figure below depicts a logic model of the Organization entity. This logic model is based on:

- IHE ITI Health Provider Directory (HPD) Profile

- ISO FDIS 21091-2012 Health informatics — Directory services for healthcare providers, subjects of care and other entities (by reference in the HPD Profile)

- HL7v3 Standard: Healthcare, Community Services and Provider Directory, Release 1
2190   (2010) – Appendix C HSD Data Dictionary
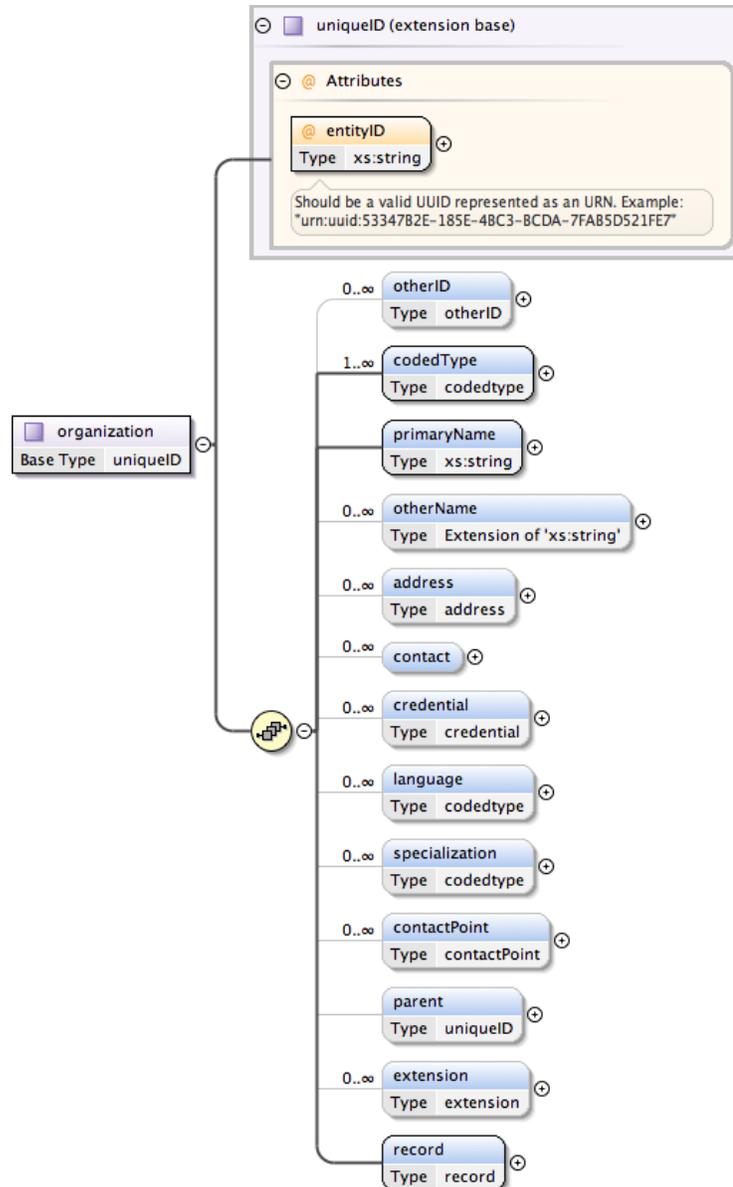
Figure Y.1-1: Organization Logical Model

Figure Y.1-1 shows the data model as it pertains to an Organization.

- An organization shall have one unique identifier, its entityID, which is conformant to
2195 RFC 4122.

- An organization may have additional other identifiers.

- An organization shall have at least one type.

- An organization shall have a status. This status may be "Active", meaning that it is currently providing or willing to provide services, or "Inactive".

2200
- An organization shall have one Primary name that is their preferred name.

- An organization may have additional names.

- An organization may have addresses (e.g., mailing address, delivery address). There may be multiple addresses.

- An organization shall indicate a contact point, if known. The contact person (including
2205 their contact details), language, and contact details (which may include phone, email, fax number, etc.) shall be indicated for the organization.

- An organization may have identified operating languages.

- An organization's specialty may be indicated.

- Organizations may have relationships to other entities. An organization may have a
2210 defined parent organization or it may, itself, be a parent to one or more sub-organizations.

The attributes of an Organization are described in Table Y.1-1.

Table Y.1-1: Organization Attributes

Concept	Optionality & Cardinality	Definition & Location Relative To `/CSD/organizationDirectory/organization`
Unique Entity Identifier	R 1..1	The globally unique identifier for this organization. `./@entityID`
Org Identifiers	O 0..∞	An organization may have multiple other identifiers. Type values will be defined by national or regional organizations. `./otherID`
Org Type	R 1..∞	The values for organization type will be defined by national or regional organizations. `./codedType/@code` The code value for the organization (required) `./codedType/@codingScheme` The coding scheme used (required)
Org Primary Name	R 1..1	The organization's preferred name. `./primaryName`

Concept	Optionality & Cardinality	Definition & Location Relative To /CSD/organizationDirectory/organization
Org other Names	O 0..∞	Other names for the organization. `./otherName`
Org Contact	R2 0..∞	A contact defined for the organization. `./contact`
Org Admin Address	R2 0..∞	The addresses for this organization. Addresses may be identified as billing, delivery, mailing, etc. `./address`
Org Credentials	O 0..∞	The list of credentials held by the organization. NOTE: degree is not a valid credential type for an Organization `./credential`
Org Language Supported	R2 0..∞	The languages that the organization is able to operate in. The languages shall be from the coding system defined by IETF BCP 47 and the coding scheme shall be "BCP 47" `./language`
Org Specialty	O 0..∞	Specialty services provided by this organization. `./specialization`
UDDI Business Entity URI	O 0..∞	Electronic address information of an Organization. `./contactPoint`

Encryption Certificate	O
	0..∞
		Organization's PKI certificate(s). On a local basis, other coding schemes may be adopted.
		`./contactPoint[`
		`@code="PKI" and`
		
		`@codingScheme="urn:ihe:iti:csd:2013:contactPoint"`
		`]`

Org Business Phone	R2
	0..∞
		The main telephone number(s) for the organization. On a local basis, other coding schemes may be adopted.
		`./contactPoint[`
		`@code="BP" and`
		
		`@codingScheme="urn:ihe:iti:csd:2013:contactPoint"`
		`]`

Org Fax	R2
	0..∞
		The main FAX number(s) for the organization. On a local basis, other coding schemes may be adopted.
		`./contactPoint[`
		`@code="FAX" and`
		
		`@codingScheme="urn:ihe:iti:csd:2013:contactPoint"`
		`]`

Concept	Optionality & Cardinality	Definition & Location Relative To `/CSD/organizationDirectory/organization`
Parent organization	O 0..1	If the organization is a suborganization, the unique IDs of the parents shall be indicated here. If the organization is the "top" level organization, its own unique ID may be indicated here. If this field is not specified, it is assumed that the organization is a top-level organization. `./parent/@entityID`
Creation Date	R 1..1	Creation timestamp of this record. `./record/@created`
Last Update Date	R 1..1	Last update timestamp of this record. `./record/@updated`
Org Status	R 1..1	The organization's status may be either Active/Inactive. Active means currently providing or willing to provide services. `./record/@status`

2215 **Facility**

Figure Y.1-2 depicts a data model of the Facility entity. This model is based on the following:

1. WHO draft guideline (2010): Creating a Master Health Facility List

2. HL7v3 Standard: Healthcare, Community Services and Provider Directory, Release 1 (2010) – Appendix C HSD Data Dictionary
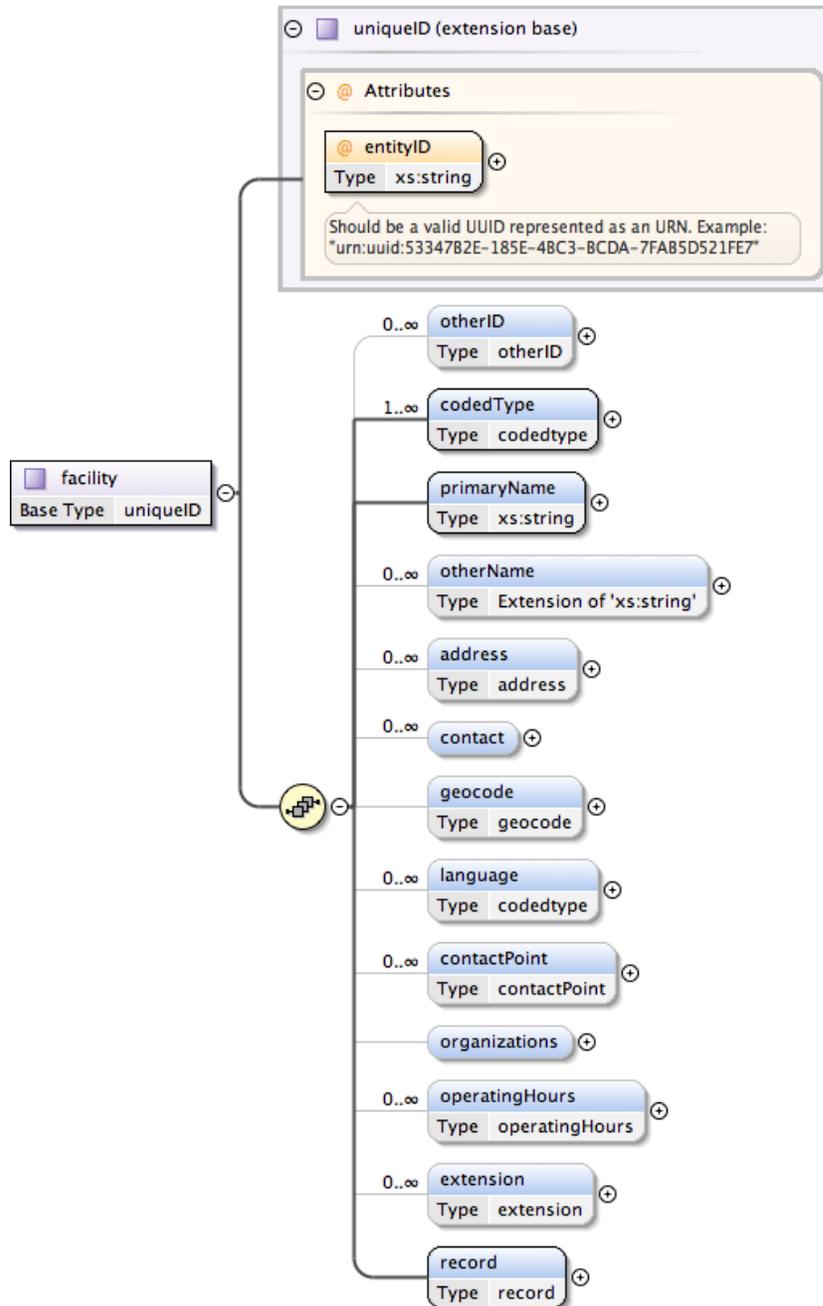
2220

Figure Y.1-2: Facility Logical Model

Facilities, as defined for the purposes of the CSD Profile, are sites from which care services are delivered.

2225
- A facility shall have one unique identifier, its entityID, which is conformant to RFC 4122.

- A facility may have additional identifiers.

- A facility shall have at least one name, there may be additional names.

- A facility shall have a defined status and at least one type

2230
- A facility may have delivery address, contact person, geocode, and language information.

- A facility may have one or more defined contact points (including, for example, a phone number, fax number, URI, PKI certificate details, etc.).

- A facility shall have a relationship with at least one organization referenced by a valid organization ID. It may have additional relationships.

2235
- A facility may have defined hours of operation including planned shut-down times (for national holidays, for example, or during periods of re-construction or repair).

- A facility may have multiple implementer-defined attributes associated with it (e.g., bed count, water point availability, internet access availability, etc.).

2240
Table Y.1-2: Facility Attributes

Concept	Optionality & Cardinality	Definition & Location Relative To `/CSD/facilityDirectory/facility/`
Unique Entity Identifier	R 1..1	The globally unique identifier for this organization. `./@entityID`
Facility Identifiers	O 0..∞	A facility may have multiple other identifiers. Type values will be defined by national or regional organizations. `./otherID`
Facility Type	R 1..∞	The values for facility type will be defined by national or regional organizations. `./codedType/@code` The code value for the facility (required) `./codedType/@codingScheme` The coding scheme used (required)
Facility Name	R 1..1	The facility's name, for primary use. `./primaryName`
Facility Known names	O 0..∞	Other optional names for the facility. `./otherName`
Facility Address	R2 0..∞	The address, if known. More than one address may be specified, but the primary address must be indicated as such. `./address`
Facility Contact	R2 0..∞	The contact defined for the facility, if known. `./contact`

Concept	Optionality & Cardinality	Definition & Location Relative To `/CSD/facilityDirectory/facility/`
Facility Geocode	R2 0..1	The facility's geocoded point position, if known. The geocode will support display of the facility's location by map/GIS applications and enable calculation of the facility's proximity to a known position. Geocodes shall be conformant to the WGS-84 specification. `./geoCode`
Facility Language Supported	R2 0..∞	The languages the facility is able to operate in, if known. The languages shall be from the coding system defined by IETF BCP 47and the coding scheme shall be "BCP 47". `./language`
Encryption Certificate	O 0..∞	Facility's PKI certificate(s). On a local basis, other coding schemes may be adopted. `./contactPoint[` `@code="PKI" and` `@codingScheme="urn:ihe:iti:csd:2013:contactPoint"` `]`
Facility Business Phone	R2 0..∞	The main telephone number(s) for the facility. On a local basis, other coding schemes may be adopted. `./contactPoint[` `@code="BP" and` `@codingScheme="urn:ihe:iti:csd:2013:contactPoint"` `]`
Facility Fax	R2 0..∞	The main FAX number(s) for the organization. On a local basis, other coding schemes may be adopted. `./contactPoint[` `@code="FAX" and` `@codingScheme="urn:ihe:iti:csd:2013:contactPoint"` `]`
Facility Org Parent	R 1..∞	The "parent" organization; there may be more than one parent organization for a facility. The ID referenced shall be a valid organizational ID in the organization directory. `./organizations/organization/@entityID`
Facility Operating Hours	R2 0..∞	The operating hours for the facility, if known. This indicates the open hours, days of the week, and any shutdown periods (e.g., civic or national holidays, etc.). These data are employed, along with FreeBusy "busy" data, to calculate schedulable time slots. `./operatingHours`

Concept	Optionality & Cardinality	Definition & Location Relative To `/CSD/facilityDirectory/facility/`
Facility Services	R2 0..∞	The care services offered at the facility, if known. There may be many services provided. The service ID, if specified, shall reference a valid ID in the service directory. Optional attributes of this relationship may be defined, such as name, language, operating hours, and freeBusy URI. See Figure Y.1-3. `./organizations/organization/service/@entityID`
Creation DateTime	R 1..1	The creation timestamp for this record. `./record/@created`
Last Update DateTime	R 1..1	The last update timestamp for this record. `./record/@updated`
Facility Status	R 1..1	The facility's status may be either Active/Inactive. Active means currently providing or willing to provide services. `./record/@status`

FacilityService

Figure Y.1-3: FacilityService Logical Model

2245

Figure Y.1-3 shows the FacilityService entity that describes the relationship between a Facility and the Services it provides. The parent organization of the FacilityService is unambiguously

established by establishing a 1:1 relationship between the service element and its parent organization element. Therefore, there is one facility/organizations/organization/service element under /CSD/facilityDirectory for each unique combination of Facility, Service and Organization. Information about these relationship attributes is described in Table Y.1-3.
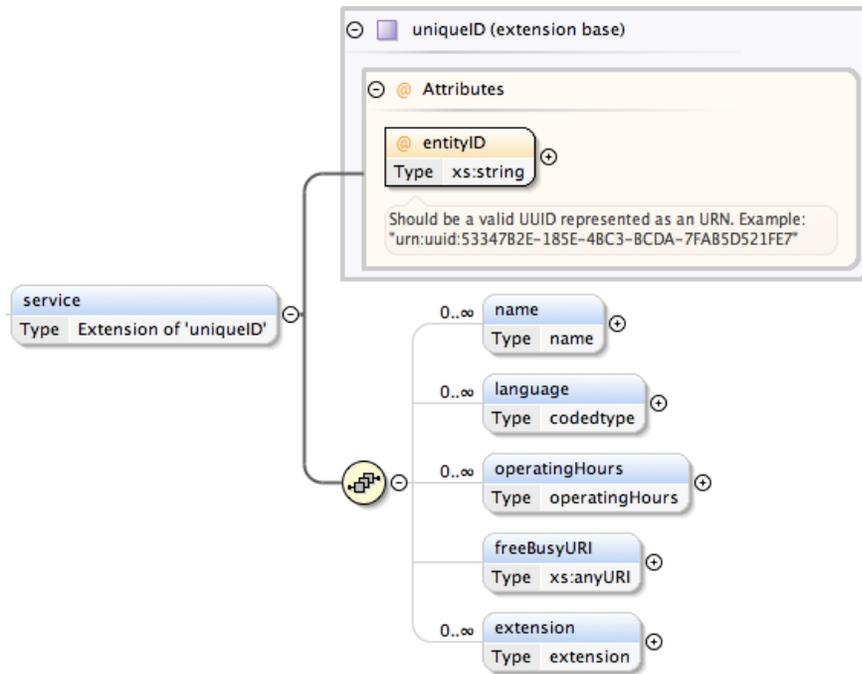
Table Y.1-3: FacilityService Attributes

Concept	Optionality & Cardinality	Definition & Location Relative To /CSD/facilityDirectory/facility/ organizations/ organization/service
Unique Entity Identifier	R 1..1	The globally unique identifier for the service offered by the organization at this facility. `./@entityID`
Name	O 0..∞	A name that may be employed to reference this FacilityService. `./name`
Language Supported	O 0..∞	The languages may be indicated for this FacilityService, if different from the facility languages,. The languages shall be from the coding system defined by IETF BCP 47and the coding scheme shall be "BCP 47" `./language`
Operating Hours	R2 0..∞	The operating hours for the FacilityService, if known. This indicates the open hours, days of the week, and any shutdown periods (e.g., civic or national holidays, etc.) for this service and may differ from the operating hours of the overall facility. These data are employed, along with FreeBusy "busy" data, to calculate schedulable time slots. `./operatingHours`
FreeBusy URI	R2 0..∞	This identifies the endpoint of the FreeBusy entity that maintains freeBusy data about this FacilityService. It shall be specified if known. `./freeBusyURI`

Provider

Figure Y.1-4 below depicts a data model of the Provider entity.
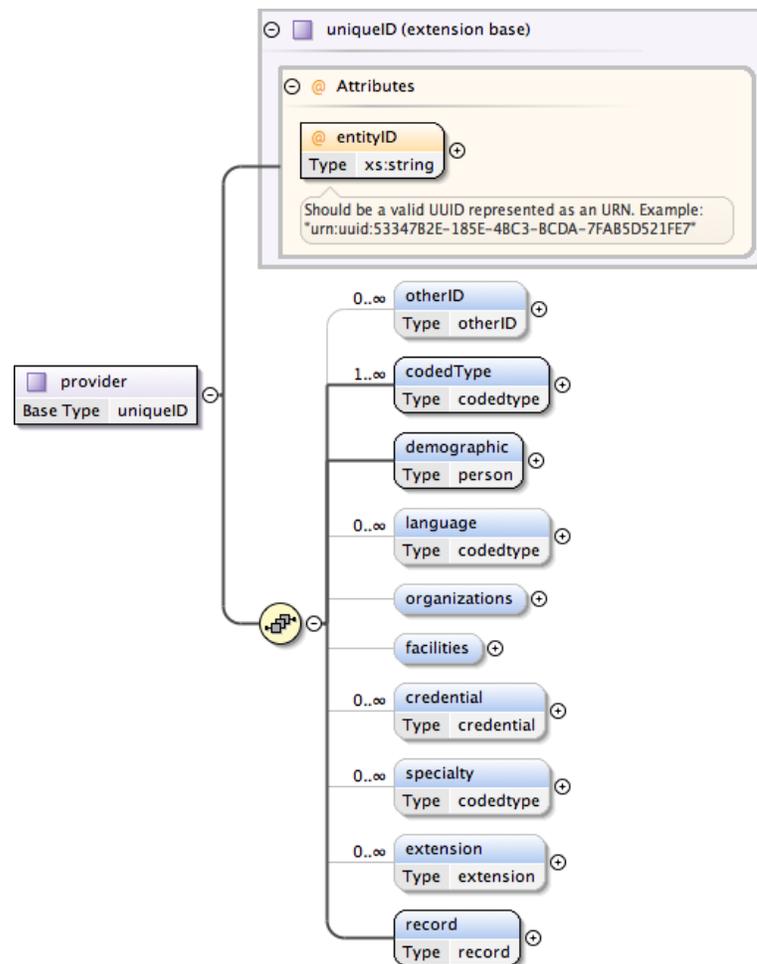
Figure Y.1-4: Provider Logical Model

2260
- A provider shall have one unique identifier, its entityID, which is conformant to RFC 2141.

- A provider may have additional identifiers.

- A provider shall have a defined status and at least one type

- A provider shall have at least one name.

2265
- A provider may be related to one or more organizations or facilities.

- A provider may have addresses (e.g., mailing address or delivery address),

- A provider may have contact details (which may include phone, email, fax number, URI, PKI certificate details, etc.),

- A provider may have one or more identified operating languages (language will be a
2270   coded value).

- A provider may have hours of operation associated with its relationships to service provision at a facility.

- A provider may have FreeBusy data associated with defined FacilityService relationships.

2275

Table Y.1-4: Provider Attributes

Concept	Optionality & Cardinality	Definition & Location Relative To `/CSD/providerDirectory/provider/`
Unique Entity Identifier	R 1..1	The globally unique identifier for this provider. `./@entityID`
Provider "Identifiers"	O 0..∞	A provider may have multiple other identifiers. Type values will be defined by national or regional organizations. `./@otherID`
Provider Type	R 1..∞	The values for provider type will be defined by national or regional organizations. An example provider type code set would be the ISO 21298 Functional Roles or the International Standard Classification of Occupations (http://www.ilo.org/public/english/bureau/stat/isco/isco08/). `./codedType/@code` The code value for the provider (required) `./codedType/@codingScheme` The coding scheme used (required)
Provider Name	R 1..∞	The provider's full name. This may be the primary search value used in discovery-by-name. `./demographic/name/commonName`
Provider Title	O 0..∞	This text field is locally defined. It can be used for "honorific job title" (e.g., Dr., Nurse, Pharmacist) or for organization positional (e.g., Head of Surgery), etc. `./demographic/name/honorific`
Provider First name	R2 0..∞	The provider's first (forename) name must be specified. `./demographic/name/forename`
Provider Middle Name	O 1..∞	The provider's middle name(s) may be specified here. `./demographic/name/otherName`
Provider Last Name	R2 0..∞	The provider's last (surname) name is a required field, if known. `./demographic/name/surname`
Provider Known names	O 0..∞	The provider's other known names may be listed here. `./demographic/name/commonName`
Provider Language Supported	R2 0..∞	The provider's supported language(s), if known The languages shall be from the coding system defined by IETF BCP 47and the coding scheme shall be "BCP 47" `./language`

Concept	Optionality & Cardinality	Definition & Location Relative To /CSD/providerDirectory/provider/
Provider Gender	R2 0..1	The provider's gender the values of which are locally defined ./demographic/gender
Encryption Certificate	O 0..∞	Facility's PKI certificate(s). On a local basis, other coding schemes may be adopted. ./demographic/contactPoint[@code="PKI" and @codingScheme="urn:ihe:iti:csd:2013:contactPoint"]
Facility Business Phone	R2 0..∞	The main telephone number(s) for the facility. On a local basis, other coding schemes may be adopted. ./demographic/contactPoint[@code="BP" and @codingScheme="urn:ihe:iti:csd:2013:contactPoint"]
Facility Fax	R2 0..∞	The main FAX number(s) for the organization. On a local basis, other coding schemes may be adopted. ./demographic/contactPoint[@code="FAX" and @codingScheme="urn:ihe:iti:csd:2013:contactPoint"]
Provider Practice Organization	O 0..∞	A provider may have a relationship with one or more organizations. ./organizations/organization/@entityID
Provider Facility	O 0..∞	A provider may have a relationship with multiple facilities from which care services are provided; these are cross-referenced here by facility ID. ./facilities/facility/@entityID
Provider Facility Service	O 0..∞	A provider may have service-delivery relationships with a facility or facilities. A reference to each Service + Facility combination is maintained; this supports discovery regarding where (and when) this provider is delivering which services. ./facilities/facility/service[@entityID]
Provider Address	R2 0..∞	The provider's address, if known. A provider may have multiple addresses of different types (e.g., mailing, billing, etc.). ./demographic/address
Provider "Credential"	O 0..∞	Specific certifications or accreditations may be listed here. ./credential
Provider Specialty	O 0..∞	A coded provider's "area of practice" may be indicated here. For discovery, this is a coded value. The code system is left up to the implementer. E.g., ISO 21298 Functional Roles: i.e., Dermatology, Oncology, Emergency Medicine. This extends/enhances the precision of the Provider Type. ./specialty

Concept	Optionality & Cardinality	Definition & Location Relative To /CSD/providerDirectory/provider/
Provider Org Address	O 0..∞	A provider addresses as determined by an organizational relationship. ./organizations/organization/address
Provider Org Contact Points	O 0..∞	A provider contact points (phone, email, fax, encryption certifications, etc.) unique to an organizational relationship. ./organizations/organization/contactPoint
Creation Date	R 1..1	The creation timestamp of this record. ./record/@created
Last Update Date	R 1..1	The last update timestamp of this record. ./record/@updated
Provider Status	R 1..1	For backward compatibility with HPD, the provider status may be either: Active/Inactive/Deceased/Retired. ./record/@status

FreeBusy

2280 FreeBusy data may be available for FacilityService or Provider entities. This data lists the "busy" time for a Provider or for a FacilityService entity within a specified timeframe. Along with the Provider or FacilityService operating hours, these data may be employed to calculate schedulable time slots.

This data is not maintained as an element for retrieval as an entity. It is retrieved using the Free Busy query [ITI-75].

2285 ### Service
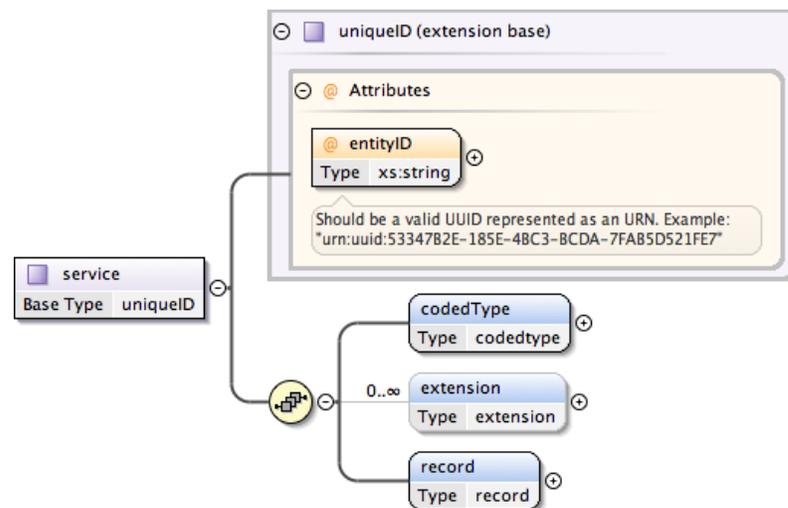
Figure Y.1-5: Service Element

The Service directory data model is shown in Figure Y.1-5 Service Element. The Service
2290 directory defines the service IDs which are employed by the Facility and Provider directories to
unambiguously describe the services provided.

- A service shall have one unique identifier, its entityID, which is conformant RFC 4122.

- Each service shall have a coded type; the type values are defined by the implementing jurisdiction.

2295 The attributes of a service are described in Table Y.1-5.

Table Y.1-5: Service Directory Attributes

Concept	Optionality & Cardinality	Definition & Location Relative To `/CSD/serviceDirectory/service/`
Unique Entity Identifier	R 1..1	The globally unique identifier for this service. `./@entityID`
Service Type	R 1..1	The values for service type will be defined by national or regional organizations. `./codedType/@code` The code value for the service (required) `./codedType/@codingScheme` The coding scheme used (required)
Status	R 1..1	The service status may be either: Active/Inactive `./record/@status`
Creation Date	R 1..1	The creation timestamp of this record. `./record/@created`
Last Update Date	R 1..1	The last update timestamp of this record. `./record/@updated`

Primitives

2300
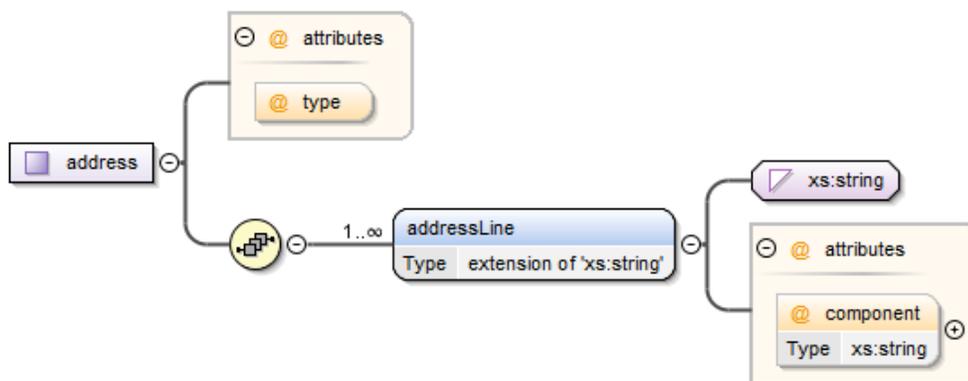
Figure Y.1-6: Address and AddressLine

The Address logic diagram is shown in Figure Y.1-6. An address may have a type, such as
Mailing, Delivery, Legal, etc. These types shall be defined by an implementing jurisdiction. Each
address shall be made up of at least 1, but possibly many, Address Lines. Each Address Line
may have a component attribute, such as StreetAddress, City, Province, Country, PostalCode,
etc. Address Line components shall be defined by an implementing jurisdiction.

An Address shall be constructed as a collection of one or more `addressLine` elements. An
example is shown below.

```xml
<address type="Legal">
  <addressLine component="StreetAddress">157 Amberly Blvd</addressLine>
  <addressLine component="City">Hamilton</addressLine>
  <addressLine component="Province">ON</addressLine>
  <addressLine component="Country">Canada</addressLine>
  <addressLine component="PostCode">L9G3V3</addressLine>
</address>
```
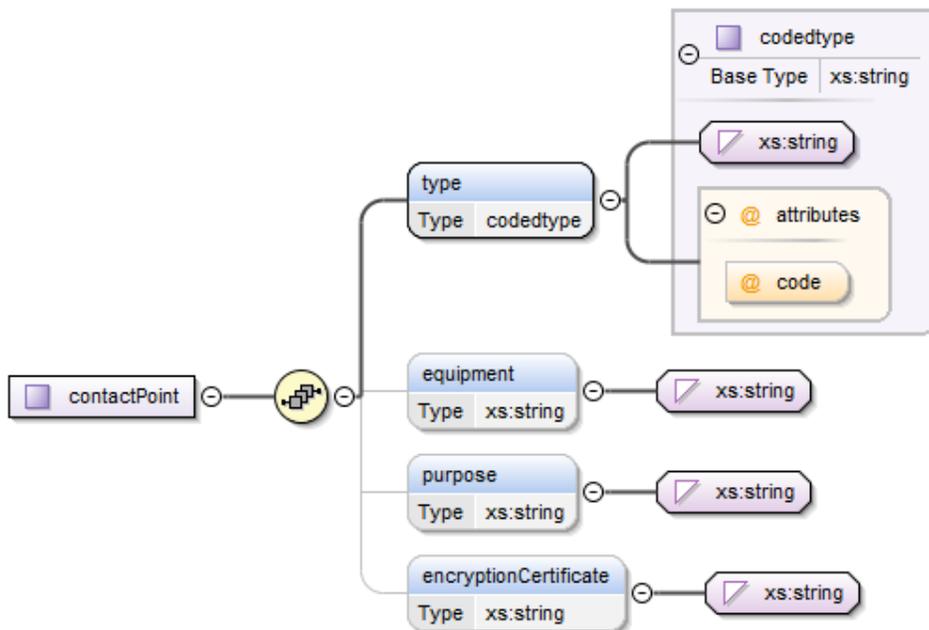

Figure Y.1-7: Contact Point

The Contact Point logic diagram is shown in Figure Y.1-7. A contact point is used to describe
communications endpoints. There may be many different kinds of contact points including
telephone, cell phone, fax, email, web service, etc. The type of the contact point shall be
identified and the coding scheme for indicating contact point types shall be defined by an

implementing jurisdiction. The equipment, purpose and encryption certificates for contact points may be optionally indicated.
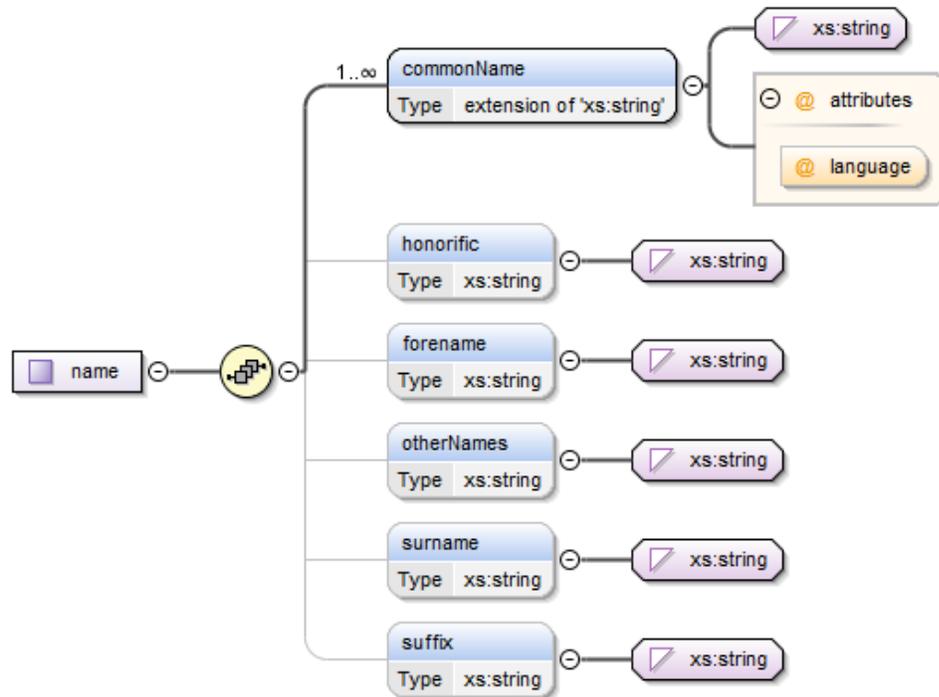
2330

Figure Y.1-8: Name Element

The logic diagram for Name is shown in Figure Y.1-8. A name shall contain at least one element, the Common Name. This element has an optional language attribute. Because of the particular challenges regarding the representation of names across multiple languages and in different cultures, attention is drawn to the ITI TF-2a: 3.24.5.2.3.1. This section in the ITI TF describes the use of the language tag as documented in BCP 47 and the HL7 XCN name data type. The CSD XSD has expressly defined the commonName element so that implementers may make use of these methods in supporting complex international conventions regarding names. The specific format for Common Name and (optional) use of the language attribute shall be defined by implementing jurisdictions.

2335

2340

Each name shall include at least the commonName element. Optionally, the honorific, forename, otherName (middle names), surname and suffix may be indicated.

Y.2 CSD Schema

The following is the normative version of the schema. For developer convenience this is extracted and provided in ITI TF-2x: Appendix W. In the event of a conflict, this normative version shall be followed.

2345

```xml
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xforms="http://www.w3.org/2002/xforms"
  elementFormDefault="qualified"
  targetNamespace="urn:ihe:iti:csd:2013"
  xmlns="urn:ihe:iti:csd:2013"
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <xs:import namespace="http://www.w3.org/2001/XMLSchema"
    schemaLocation="http://www.w3.org/2001/XMLSchema.xsd"/>
  <xs:import namespace="http://www.w3.org/2002/xforms"
    schemaLocation="http://www.w3.org/MarkUp/Forms/2002/XForms-Schema.xsd"/>
  <xs:import namespace = "http://www.w3.org/XML/1998/namespace"/>


  <xs:complexType name="uniqueID">
    <xs:attribute name="entityID" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="otherID">
    <xs:attribute name="code" type="xs:string" use="required"/>
    <xs:attribute name="assigningAuthorityName" type="xs:string" use="required">
      <xs:annotation>
        <xs:documentation>Use of an OID is encouraged.
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>


  <xs:complexType name="extension">
    <xs:sequence>
      <xs:any minOccurs="0" namespace="##any" processContents="skip"/>
    </xs:sequence>
    <xs:attribute name="type" use="required"> </xs:attribute>
    <xs:attribute name="urn" type="xs:string" use="required"> </xs:attribute>
    <xs:anyAttribute/>
  </xs:complexType>


  <xs:complexType name="codedtype">
    <xs:annotation>
      <xs:documentation>The text content of this element may optionally be used to provide
translations to the local language of a coded value.</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="code" use="required"/>
        <xs:attribute name="codingScheme" use="required"/>
        <xs:anyAttribute/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

```
     <xs:complexType name="record">
       <xs:attribute name="created" type="xs:dateTime" use="required"/>
2405   <xs:attribute name="updated" type="xs:dateTime" use="required"/>
       <xs:attribute name="status" type="xs:string" use="required"/>
       <xs:attribute name="sourceDirectory" type="xs:anyURI"/>
     </xs:complexType>

2410 <!--intended to be compatible with HPD information model)-->
     <xs:complexType name="organization">
       <xs:complexContent>
         <xs:extension base="uniqueID">
           <xs:sequence>
2415         <xs:element name="otherID" type="otherID" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="codedType" type="codedtype" maxOccurs="unbounded"/>
             <xs:element name="primaryName" type="xs:string"/>
             <xs:element name="otherName" minOccurs="0" maxOccurs="unbounded">
               <xs:complexType>
2420             <xs:simpleContent>
                   <xs:extension base="xs:string">
                     <xs:attribute ref="xml:lang"/>
                   </xs:extension>
                 </xs:simpleContent>
2425           </xs:complexType>
             </xs:element>
             <xs:element name="address" type="address" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element maxOccurs="unbounded" minOccurs="0" name="contact">
               <xs:complexType>
2430             <xs:choice>
                   <xs:element name="provider" type="uniqueID"/>
                   <xs:element name="person" type="person"/>
                 </xs:choice>
               </xs:complexType>
2435         </xs:element>
             <xs:element name="credential" type="credential" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="language" type="codedtype" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="specialization" type="codedtype" minOccurs="0"
maxOccurs="unbounded"/>
2440         <xs:element name="contactPoint" type="contactPoint" maxOccurs="unbounded"
minOccurs="0"/>
             <xs:element name="parent" type="uniqueID" minOccurs="0"/>
             <xs:element minOccurs="0" maxOccurs="unbounded" name="extension" type="extension"/>
             <xs:element name="record" type="record"/>
2445       </xs:sequence>
         </xs:extension>
       </xs:complexContent>
     </xs:complexType>

2450 <!--intended to be compatible with WHO MFL draft and HL7-OMG HSD data dictionary; modified per
recommendations from openHIE FR community)-->
     <xs:complexType name="facility">
       <xs:complexContent>
         <xs:extension base="uniqueID">
2455       <xs:sequence>
             <xs:element name="otherID" type="otherID" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="codedType" type="codedtype" maxOccurs="unbounded"/>
             <xs:element name="primaryName" type="xs:string"/>
             <xs:element name="otherName" minOccurs="0" maxOccurs="unbounded">
2460           <xs:complexType>
                 <xs:simpleContent>
                   <xs:extension base="xs:string">
                     <xs:attribute ref="xml:lang"/>
                   </xs:extension>
2465             </xs:simpleContent>
               </xs:complexType>
             </xs:element>
             <xs:element name="address" type="address" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element maxOccurs="unbounded" minOccurs="0" name="contact">
```

```
2470            <xs:complexType>
                  <xs:choice>
                    <xs:element name="provider" type="uniqueID"/>
                    <xs:element name="person" type="person"/>
                  </xs:choice>
2475            </xs:complexType>
              </xs:element>
              <xs:element name="geocode" type="geocode" minOccurs="0"/>
              <xs:element name="language" type="codedtype" minOccurs="0" maxOccurs="unbounded"/>
              <xs:element name="contactPoint" type="contactPoint" maxOccurs="unbounded"
2480   minOccurs="0"/>
              <xs:element minOccurs="0" name="organizations">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="organization" maxOccurs="unbounded">
2485                  <xs:complexType>
                        <xs:complexContent>
                          <xs:extension base="uniqueID">
                            <xs:sequence>
                              <xs:element name="service" maxOccurs="unbounded" minOccurs="0">
2490                            <xs:complexType>
                                  <xs:complexContent>
                                    <xs:extension base="uniqueID">
                                      <xs:sequence>
                                        <xs:element name="name" type="name" minOccurs="0"
2495                                      maxOccurs="unbounded"/>
                                        <xs:element name="language" type="codedtype" minOccurs="0"
                                          maxOccurs="unbounded"/>
                                        <xs:element name="operatingHours" type="operatingHours"
                                          minOccurs="0" maxOccurs="unbounded"/>
2500                                    <xs:element name="freeBusyURI" minOccurs="0"
       type="xs:anyURI"> </xs:element>
                                        <xs:element minOccurs="0" maxOccurs="unbounded"
       name="extension"
                                          type="extension"/>
2505                                  </xs:sequence>
                                    </xs:extension>
                                  </xs:complexContent>
                                </xs:complexType>
                              </xs:element>
2510                          <xs:element minOccurs="0" maxOccurs="unbounded" name="extension"
                                type="extension"/>
                            </xs:sequence>
                          </xs:extension>
                        </xs:complexContent>
2515                    </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
              </xs:element>
2520          <xs:element name="operatingHours" type="operatingHours" minOccurs="0"
                maxOccurs="unbounded"/>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="extension" type="extension"/>
              <xs:element name="record" type="record"/>
            </xs:sequence>
2525      </xs:extension>
        </xs:complexContent>
      </xs:complexType>

      <!--intended to be compatible with information model from HPD; modified based on suggestions
2530  from openHIE PR community)-->
      <xs:complexType name="provider">
        <xs:complexContent>
          <xs:extension base="uniqueID">
            <xs:sequence>
2535          <xs:element name="otherID" type="otherID" minOccurs="0" maxOccurs="unbounded"/>
              <xs:element name="codedType" type="codedtype" maxOccurs="unbounded"/>
```

```
                    <xs:element name="demographic" type="person"/>
                    <xs:element name="language" type="codedtype" maxOccurs="unbounded" minOccurs="0"/>
                    <xs:element name="organizations" minOccurs="0">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="organization" maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:complexContent>
                                <xs:extension base="uniqueID">
                                  <xs:sequence>
                                    <xs:element minOccurs="0" maxOccurs="unbounded" name="extension"
                                      type="extension"/>
                                    <xs:element name="address" type="address" minOccurs="0"
                                      maxOccurs="unbounded"/>
                                    <xs:element name="contactPoint" type="contactPoint" minOccurs="0"
                                      maxOccurs="unbounded"/>
                                  </xs:sequence>
                                </xs:extension>
                              </xs:complexContent>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="facilities" minOccurs="0">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="facility" maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:complexContent>
                                <xs:extension base="uniqueID">
                                  <xs:sequence>
                                    <xs:element name="service" maxOccurs="unbounded" minOccurs="0">
                                      <xs:complexType>
                                        <xs:complexContent>
                                          <xs:extension base="uniqueID">
                                            <xs:sequence>
                                              <xs:element name="name" type="name" minOccurs="0"
                                                maxOccurs="unbounded"/>
                                              <xs:element minOccurs="0" name="organization"
type="uniqueID"/>
                                              <xs:element name="language" type="codedtype" minOccurs="0"
                                                maxOccurs="unbounded"/>
                                              <xs:element name="operatingHours" type="operatingHours"
                                                minOccurs="0" maxOccurs="unbounded"/>
                                              <xs:element name="freeBusyURI" minOccurs="0"
type="xs:anyURI"> </xs:element>
                                              <xs:element minOccurs="0" maxOccurs="unbounded"
name="extension"
                                                type="extension"/>
                                            </xs:sequence>
                                          </xs:extension>
                                        </xs:complexContent>
                                      </xs:complexType>
                                    </xs:element>
                                    <xs:element name="operatingHours" type="operatingHours" minOccurs="0"
maxOccurs="unbounded"/>
                                    <xs:element minOccurs="0" maxOccurs="unbounded" name="extension"
                                      type="extension"/>
                                  </xs:sequence>
                                </xs:extension>
                              </xs:complexContent>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
```

```
                    <xs:element name="credential" type="credential" minOccurs="0" maxOccurs="unbounded"/>
2605                <xs:element name="specialty" type="codedtype" minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element minOccurs="0" maxOccurs="unbounded" name="extension" type="extension"/>
                    <xs:element name="record" type="record"/>
                </xs:sequence>
            </xs:extension>
2610        </xs:complexContent>
        </xs:complexType>


        <!--informed by government specification from France and by OASIS names spec:
2615    http://xml.coverpages.org/namesAndAddresses.html)-->
        <xs:complexType name="name">
            <xs:sequence>
                <xs:element name="commonName" maxOccurs="unbounded">
                    <xs:complexType>
2620                    <xs:simpleContent>
                            <xs:extension base="xs:string">
                            </xs:extension>
                        </xs:simpleContent>
                    </xs:complexType>
2625                </xs:element>
                <xs:element name="honorific" type="xs:string" minOccurs="0"/>
                <xs:element name="forename" type="xs:string" minOccurs="0"/>
                <xs:element name="otherName" type="codedtype" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="surname" type="xs:string" minOccurs="0"/>
2630            <xs:element name="suffix" type="xs:string" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>

        <xs:complexType name="person">
2635        <xs:sequence>
                <xs:element name="name" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:complexContent>
                            <xs:extension base="name">
2640                            <xs:attribute ref="xml:lang"/>
                            </xs:extension>
                        </xs:complexContent>
                    </xs:complexType>
                </xs:element>
2645            <xs:element name="contactPoint" type="contactPoint" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="address" type="address" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="gender" type="xs:string" minOccurs="0"/>
                <xs:element minOccurs="0" name="dateOfBirth" type="xs:date"/>
                <xs:element minOccurs="0" maxOccurs="unbounded" name="extension" type="extension"/>
2650        </xs:sequence>
        </xs:complexType>

        <!--ref HL7-OMG HSD data dictionary and IHE HPD)-->
        <xs:complexType name="contactPoint">
2655        <xs:sequence>
                <xs:element name="codedType" type="codedtype"/>
                <xs:element name="equipment" type="xs:string" minOccurs="0"/>
                <xs:element name="purpose" type="xs:string" minOccurs="0"/>
                <xs:element name="certificate" type="xs:string" minOccurs="0"/>
2660        </xs:sequence>
        </xs:complexType>
```

```
        <!--informed by OASIS names and addresses spec:
2665  http://xml.coverpages.org/namesAndAddresses.html)-->
        <xs:complexType name="address">
          <xs:sequence>
            <xs:element name="addressLine" maxOccurs="unbounded">
              <xs:complexType>
2670          <xs:simpleContent>
                  <xs:extension base="xs:string">
                    <!--could model extensions using address type attributes; these types could be
      defined by implementing jurisdiction; e.g., village, sector, etc.)-->
                    <xs:attribute name="component" type="xs:string" use="optional"/>
2675            </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
2680      <xs:attribute name="type"/>
        </xs:complexType>

        <!--intended to be backward compatible with HPD and informed by HL7-OMG HSD spec)-->
        <xs:complexType name="credential">
2685      <xs:sequence>
            <xs:element name="codedType" type="codedtype"/>
            <xs:element name="number" type="xs:string"/>
            <xs:element name="issuingAuthority" type="xs:string" minOccurs="0"/>
            <xs:element name="credentialIssueDate" type="xs:date" minOccurs="0"/>
2690        <xs:element name="credentialRenewalDate" type="xs:date" minOccurs="0"/>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="extension" type="extension"/>
          </xs:sequence>
        </xs:complexType>

2695    <xs:complexType name="geocode">
          <xs:sequence>
            <xs:element name="latitude" type="xs:decimal"/>
            <xs:element name="longitude" type="xs:decimal"/>
            <xs:element name="altitude" type="xs:decimal" minOccurs="0"/>
2700        <xs:element name="coordinateSystem" type="xs:string" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>

        <!--informed by HL7-OMG HSD spec; Sunday=0, Monday=1...)-->
2705    <xs:complexType name="operatingHours">
          <xs:sequence>
            <xs:element name="openFlag" type="xs:boolean"/>
            <xs:element name="dayOfTheWeek" type="xs:integer" maxOccurs="unbounded" minOccurs="0"/>
            <xs:element name="beginningHour" type="xs:time" minOccurs="0"/>
2710        <xs:element name="endingHour" type="xs:time" minOccurs="0"/>
            <xs:element name="beginEffectiveDate" type="xs:date"/>
            <xs:element name="endEffectiveDate" type="xs:date" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
2715
        <!--informed by HL7-OMG HSD spec)-->
        <xs:complexType name="service">
          <xs:complexContent>
            <xs:extension base="uniqueID">
2720          <xs:sequence>
                <xs:element name="otherID" type="otherID" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="codedType" type="codedtype" maxOccurs="unbounded"/>
                <xs:element minOccurs="0" maxOccurs="unbounded" name="extension" type="extension"/>
                <xs:element name="record" type="record"/>
2725          </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
```

```
2730      <xs:complexType name="CSD">
            <xs:sequence>
              <xs:element name="organizationDirectory">
                <xs:complexType>
2735              <xs:sequence>
                    <xs:element name="organization" type="organization" minOccurs="0"
maxOccurs="unbounded"
                    />
                  </xs:sequence>
2740            </xs:complexType>
              </xs:element>
              <xs:element name="serviceDirectory">
                <xs:complexType>
                  <xs:sequence>
2745                <xs:element name="service" type="service" minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="facilityDirectory">
2750            <xs:complexType>
                  <xs:sequence>
                    <xs:element name="facility" type="facility" minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
2755          </xs:element>
              <xs:element name="providerDirectory">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="provider" type="provider" minOccurs="0" maxOccurs="unbounded"/>
2760              </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
2765
          <xs:element name="CSD" type="CSD"/>


2770      <xs:element name="careServicesRequest">
            <xs:complexType>
                <xs:sequence>
                  <xs:any minOccurs="0" namespace="##any" processContents="skip"/>
                </xs:sequence>
2775            <xs:attribute name="id" />
                <xs:attribute name="uuid" use="required" type="uuid">
                  <xs:annotation>
                    <xs:documentation>Refers to a UUID identifying a &lt;careServicesFunction/&gt; to
be processed by the Care Services InfoManager. These stored queries must be defined in a
2780 &lt;careServicesFunction/&gt;.</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
                <xs:anyAttribute>
                  <xs:annotation>
2785                <xs:documentation>Extension point for implementation specific
attributes</xs:documentation>
                  </xs:annotation>
                </xs:anyAttribute>
            </xs:complexType>
2790      </xs:element>
          <xs:element name="careServicesResponse">
            <xs:complexType>
              <xs:sequence>
```

```
2795            <xs:element name="result" type="xs:anyType"/>
          </xs:sequence>
          <xs:attribute name="content-type" type="xs:string">
            <xs:annotation>
              <xs:documentation>The content or mime-type of the query result.

2800  Values must be conformant to
       http://tools.ietf.org/html/rfc2045
       http://tools.ietf.org/html/rfc2046</xs:documentation>
            </xs:annotation>
          </xs:attribute>
2805      <xs:attribute name="code" type="xs:integer"/>
          <xs:attribute name="id" type="xs:string"/>
          <xs:anyAttribute>
            <xs:annotation>
              <xs:documentation>Extension point for implementation specific
2810  attributes</xs:documentation>
            </xs:annotation>
          </xs:anyAttribute>
        </xs:complexType>
      </xs:element>
2815    <xs:element name="careServicesFunction">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="description" minOccurs="0" type="xs:anyType">
              <xs:annotation>
2820            <xs:documentation>An optional description of what this query performs
                </xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="definition" type="xs:string">
2825            <xs:annotation>
                <xs:documentation>The xquery which defines the result that the Care Services
      InfoManager will return upon request by the Service Finder to execute the query.

      The function shall be executed against the root node of the CSD document managed by the Care
2830   Services InfoManager. Any parameters passed to the function are accessible in the dynamic
       context of the function under the variable $careServicesRequest.
                </xs:documentation>
              </xs:annotation>
            </xs:element>
```

```
2835            <xs:element ref="xforms:instance" minOccurs="0"/>
                <xs:element ref="xs:schema" minOccurs="0" maxOccurs="1"/>
                <xs:element ref="xforms:bind" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="extension" type="extension"/>
            </xs:sequence>
2840        <xs:attribute name="uuid" use="required" type="uuid">
              <xs:annotation>
                <xs:documentation>UUID used to identify the stored function, in particular by the
Service Finder. Must conform to RFC 4122 http://www.ietf.org/rfc/rfc4122.txt</xs:documentation>
              </xs:annotation>
2845        </xs:attribute>
            <xs:attribute name="content-type" type="xs:string" use="required">
              <xs:annotation>
                <xs:documentation>The content or mime-type of the query result.
Values must be conformant to:
2850   http://tools.ietf.org/html/rfc2045
       http://tools.ietf.org/html/rfc2046</xs:documentation>
              </xs:annotation>
            </xs:attribute>
            <xs:attribute name="schema" type="xforms:anyURIList" use="optional"/>
2855        <xs:anyAttribute>
              <xs:annotation>
                <xs:documentation>Extension point for implementation specific
attributes</xs:documentation>
              </xs:annotation>
2860        </xs:anyAttribute>
          </xs:complexType>
        </xs:element>
     </xs:schema>
```

Y.3 Stored Queries

This is the XML definition of the `csd:careServicesRequest` for Organization Search. A non-normative implementation of the referenced organization_search.xq is included in ITI TF-2x: Appendix W.

```
2870   <?xml version="1.0" encoding="UTF-8"?>
       <csd:careServicesFunction
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:csd="urn:ihe:iti:csd:2013"
         xmlns:xforms="http://www.w3.org/2002/xforms"
         xmlns:xi="http://www.w3.org/2001/XInclude"
2875     xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
         xmlns:wsa="http://www.w3.org/2005/08/addressing"
         xmlns:xsd="http://www.w3.org/2001/XMLSchema"
         xmlns:http="http://expath.org/ns/http-client"
         xmlns:ev="http://www.w3.org/2001/xml-events"
2880     urn="urn:ihe:iti:csd:2014:stored-function:organization-search">
         <csd:description>
           Performs a search for all facilities by name, type, address or ID.
         </csd:description>
         <csd:definition ><xi:include parse='text' href='organization_search.xq'/></csd:definition>
2885
         <xforms:model id='urn:ihe:iti:csd:2014:stored-function:organization-search'>
           <xforms:instance id="care-services-result">
            <csd:careSerivcesResponse id="" code="" content-type=""/>
           </xforms:instance>
2890
           <xforms:instance id='soap-request'>
            <soap:Envelope >
             <soap:Header>
               <wsa:Action soap:mustUnderstand="1">
2895             urn:ihe:iti:csd:2013:GetCareServicesRequest
```

```
        </wsa:Action>
        <wsa:MessageID/>
        <wsa:ReplyTo soap:mustUnderstand="1">
         <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
2900    </wsa:ReplyTo>
        <wsa:To soap:mustUnderstand="1">{$LOCATION}</wsa:To>
       </soap:Header>
       <soap:Body>
        <csd:careServicesRequest
2905       urn="urn:ihe:iti:csd:2014:stored-function:organization-search">
         <csd:requestParams>
          <csd:id/>
          <csd:otherID/>
          <csd:primaryName/>
2910      <csd:name/>
          <csd:codedType/>
          <csd:address/>
          <csd:parent/>
          <csd:start/>
2915      <csd:max/>
          <csd:record/>
         </csd:requestParams>
        </csd:careServicesRequest>
       </soap:Body>
2920   </soap:Envelope>
      </xforms:instance>

      <xforms:instance id='http-post-request'>
       <csd:requestParams>
2925    <csd:id/>
        <csd:otherID/>
        <csd:primaryName/>
        <csd:name/>
        <csd:codedType/>
2930    <csd:address/>
        <csd:parent/>
        <csd:start/>
        <csd:max/>
        <csd:record/>
2935   </csd:requestParams>
      </xforms:instance>

      <xforms:instance id="soap-response" >
       <soap:Envelope>
2940    <soap:Header>
         <wsa:Action soap:mustUnderstand="1">
          urn:ihe:iti:csd:2013:GetCareServicesResponse
         </wsa:Action>
         <wsa:MessageID/>
2945     <wsa:ReplyTo soap:mustUnderstand="1">
          <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
         </wsa:ReplyTo>
         <wsa:RelatesTo/>
        </soap:Header>
2950    <soap:Body>
         <csd:careServicesResponse>
          <csd:result/>
         </csd:careServicesResponse>
        </soap:Body>
2955   </soap:Envelope>
      </xforms:instance>

      <xforms:instance id="http-post-response" >
       <http:result status="">
2960    <http:header name="X-CSD-Transaction-ID" value=""/>
        <http:body/>
```

```
      </http:result>
    </xforms:instance>


2965

    <xforms:submission id="soap-submission"
      method="post" mediatype="application/soap+xml"
      ref="instance('soap-request')" resource="{$LOCATION}"/>

2970  <xforms:submission id="http-post-submission"
      method="post" mediatype="text/xml"
      ref="instance('http-post-request')"
      resource="{$LOCATION}/urn:ihe:iti:csd:2014:stored-function:organization-search" />

2975

    <xforms:action ev:event="xforms-submit-done" id="http-post-prepare">
      <xforms:insert
        context="instance('http-post-response')/http:result/http:body"
        origin="instance('care-services-result')/csd:careServicesResponse/csd:result/*"/>
2980  <xforms:setValue
        bind="instance('http-post-response')/http:result/@status"
        value="instance('care-services-result')/csd:careServicesResponse/@code"/>
      <xforms:setValue
        bind="instance('http-post-response')/http:body/@content-type"
2985    value="instance('care-services-result')/csd:careServicesResponse/@content-type"/>
      <xforms:setValue
        bind="instance('http-post-response')/http:result/http:header[@name=
'X-CSD-Transaction-ID']/@value"
        value="instance('care-services-result')/csd:careServicesResponse/@id"/>
2990  </xforms:action>

    <xforms:action ev:event="xforms-submit-done" id="soap-prepare">
      <xforms:insert
        context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
2995 @content-type"
        origin="instance('care-services-result')/csd:careServicesRepsonse/@content-type"/>
      <xforms:insert
        context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
@code"
3000    origin="instance('care-services-result')/csd:careServicesRepsonse/@code"/>
      <xforms:setValue
        bind="instance('soap-response')soap:Envelope/soap:Header/wsa:messageID"
        value="instance('soap-request')csd:careServicesRequest/@id"/>
      <xforms:setValue
3005    bind="instance('soap-response')soap:Envelope/soap:Header/wsa:relatesTo"
        value="instance('soap-request')soap:Envelope/soap:Header/wsa:messageID"/>
    </xforms:action>


3010  <xforms:bind
        nodeset="instance('soap-request')/soap:Envelope:soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:id"
        type="csd:uniqueID" />
      <xforms:bind
3015    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:otherID"
        type="csd:otherID" />
      <xforms:bind
        nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
3020 csd:careServicesRequest/csd:requestParams/csd:primaryName"
        type="xs:string" />
      <xforms:bind
        nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:name"
3025    type="xs:string" />
      <xforms:bind
        nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
```

```
csd:careServicesRequest/csd:requestParams/csd:codedType"
    type="csd:codedtype" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:address"
    type="csd:address" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:parent"
    type="csd:organization" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:start"
    type="xs:int" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:max"
    type="xs:int" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:record"
    type="csd:record" />


  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:id"
    type="csd:uniqueID" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:otherID"
    type="csd:otherID" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:primaryName"
    type="xs:string" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:name"
    type="xs:string" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:codedType"
    type="csd:codedtype" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:address"
    type="csd:address" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:parent"
    type="csd:parent" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:start"
    type="xs:int" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:max"
    type="xs:int" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:record"
    type="csd:record" />

  <xforms:bind nodeset="instance('http-post-response')/csd:CSD" type="csd:CSD" />
  <xforms:bind nodeset="instance('soap-response')/soap:Envelope/soap:Body/
csd:careServicesResponse/csd:result/csd:CSD"
    type="csd:CSD" />

 </xforms:model>
</csd:careServicesFunction>
```

Figure Y.3-1: Organization Search Care Services Function

This is the XML definition of the `csd:careServicesRequest` for Provider Search. A non-normative implementation of the referenced provider_search.xq is included in ITI TF-2x: Appendix W.

```
<?xml version="1.0" encoding="UTF-8"?>
<csd:careServicesFunction
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:csd="urn:ihe:iti:csd:2013"
xmlns:xforms="http://www.w3.org/2002/xforms"
xmlns:xi="http://www.w3.org/2001/XInclude"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:http="http://expath.org/ns/http-client"
xmlns:ev="http://www.w3.org/2001/xml-events"
urn="urn:ihe:iti:csd:2014:stored-function:provider-search">
<csd:description>
  Performs a search for all providers.
</csd:description>
<csd:definition ><xi:include parse='text' href='provider_search.xq'/></csd:definition>

<xforms:model id='urn:ihe:iti:csd:2014:stored-function:provider-search'>
 <xforms:instance id="care-services-result">
  <csd:careSerivcesResponse id="" code="" content-type=""/>
 </xforms:instance>

 <xforms:instance id='soap-request'>
  <soap:Envelope >
   <soap:Header>
    <wsa:Action soap:mustUnderstand="1">
     urn:ihe:iti:csd:2013:GetCareServicesRequest
    </wsa:Action>
    <wsa:MessageID/>
    <wsa:ReplyTo soap:mustUnderstand="1">
     <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soap:mustUnderstand="1">{$LOCATION}</wsa:To>
   </soap:Header>
   <soap:Body>
    <csd:careServicesRequest urn="urn:ihe:iti:csd:2014:stored-function:provider-search">
     <csd:requestParams>
      <csd:id/>
      <csd:otherID/>
      <csd:commonName/>
      <csd:codedType/>
      <csd:address/>
      <csd:organizations/>
      <csd:facilities/>
      <csd:start/>
      <csd:max/>
      <csd:record/>
     </csd:requestParams>
    </csd:careServicesRequest>
   </soap:Body>
  </soap:Envelope>
 </xforms:instance>

 <xforms:instance id='http-post-request'>
  <csd:requestParams>
   <csd:id/>
   <csd:otherID/>
   <csd:commonName/>
   <csd:codedType/>
   <csd:address/>
   <csd:organizations/>
   <csd:facilities/>
   <csd:start/>
   <csd:max/>
   <csd:record/>
  </csd:requestParams>
 </xforms:instance>
```

```
        <xforms:instance id="soap-response" >
         <soap:Envelope>
          <soap:Header>
           <wsa:Action soap:mustUnderstand="1">
3160        urn:ihe:iti:csd:2013:GetCareServicesResponse
           </wsa:Action>
           <wsa:MessageID/>
           <wsa:ReplyTo soap:mustUnderstand="1">
            <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
3165       </wsa:ReplyTo>
           <wsa:RelatesTo/>
          </soap:Header>
          <soap:Body>
           <csd:careServicesResponse>
3170        <csd:result/>
           </csd:careServicesResponse>
          </soap:Body>
         </soap:Envelope>
        </xforms:instance>
3175
        <xforms:instance id="http-post-response" >
         <http:result status="">
          <http:header name="X-CSD-Transaction-ID" value=""/>
          <http:body/>
3180     </http:result>
        </xforms:instance>


        <xforms:submission id="soap-submission"
3185      method="post" mediatype="application/soap+xml"
          ref="instance('soap-request')" resource="{$LOCATION}"/>

        <xforms:submission id="http-post-submission"
          method="post" mediatype="text/xml"
3190      ref="instance('http-post-request')"
          resource="{$LOCATION}/urn:ihe:iti:csd:2014:stored-function:provider-search" />


        <xforms:action ev:event="xforms-submit-done" id="http-post-prepare">
3195     <xforms:insert
            context="instance('http-post-response')/http:result/http:body"
            origin="instance('care-services-result')/csd:careServicesResponse/csd:result/*"/>
          <xforms:setValue
            bind="instance('http-post-response')/http:result/@status"
3200        value="instance('care-services-result')/csd:careServicesResponse/@code"/>
          <xforms:setValue
            bind="instance('http-post-response')/http:body/@content-type"
            value="instance('care-services-result')/csd:careServicesResponse/@content-type"/>
          <xforms:setValue
3205        bind="instance('http-post-response')/http:result/http:header[@name=
        'X-CSD-Transaction-ID']/@value"
            value="instance('care-services-result')/csd:careServicesResponse/@id"/>
          </xforms:action>

3210    <xforms:action ev:event="xforms-submit-done" id="soap-prepare">
          <xforms:insert
            context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
        @content-type"
            origin="instance('care-services-result')/csd:careServicesRepsonse/@content-type"/>
3215      <xforms:insert
            context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
        @code"
            origin="instance('care-services-result')/csd:careServicesRepsonse/@code"/>
          <xforms:setValue
3220        bind="instance('soap-response')soap:Envelope/soap:Header/wsa:messageID"
            value="instance('soap-request')csd:careServicesRequest/@id"/>
```

```
    <xforms:setValue
      bind="instance('soap-response')soap:Envelope/soap:Header/wsa:relatesTo"
      value="instance('soap-request')soap:Envelope/soap:Header/wsa:messageID"/>
  </xforms:action>


  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope:soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:id"
      type="csd:uniqueID" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:otherID"
      type="csd:otherID" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:commonName"
      type="xs:string" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:codedType"
      type="csd:codedtype" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:address"
      type="csd:address" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:start"
      type="xs:int" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:max"
      type="xs:int" />
  <xforms:bind
    nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:record"
      type="csd:record" />


  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:id"
    type="csd:uniqueID" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:otherID"
    type="csd:otherID" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:commonName"
    type="xs:string" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:codedType"
    type="csd:codedtype" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:address"
    type="csd:address" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:start"
    type="xs:int" />
  <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:record"
    type="csd:record" />

  <xforms:bind nodeset="instance('http-post-response')/csd:CSD" type="csd:CSD" />

  <xforms:bind nodeset="instance('soap-response')/soap:Envelope/soap:Body/
csd:careServicesResponse/csd:result/csd:CSD"
      type="csd:CSD" />

 </xforms:model>
</csd:careServicesFunction>
```

Figure Y.3-2: Provider Search Care Services Function

This is the XML definition of the `csd:careServicesRequest` for Facility Search. A non-normative implementation of the referenced facility_search.xq is included in ITI TF-2x: Appendix W.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<csd:careServicesFunction
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:csd="urn:ihe:iti:csd:2013"
 xmlns:xforms="http://www.w3.org/2002/xforms"
 xmlns:xi="http://www.w3.org/2001/XInclude"
 xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:wsa="http://www.w3.org/2005/08/addressing"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:http="http://expath.org/ns/http-client"
 xmlns:ev="http://www.w3.org/2001/xml-events"
 urn="urn:ihe:iti:csd:2014:stored-function:facility-search">
<csd:description>
  Performs a search for all facilities by name, type, address or ID.
</csd:description>
<csd:definition ><xi:include parse='text' href='facility_search.xq'/></csd:definition>

<xforms:model id='urn:ihe:iti:csd:2014:stored-function:facility-search'>
 <xforms:instance id="care-services-result">
  <csd:careSerivcesResponse id="" code="" content-type=""/>
 </xforms:instance>

 <xforms:instance id='soap-request'>
  <soap:Envelope >
   <soap:Header>
    <wsa:Action soap:mustUnderstand="1">
     urn:ihe:iti:csd:2013:GetCareServicesRequest
    </wsa:Action>
    <wsa:MessageID/>
    <wsa:ReplyTo soap:mustUnderstand="1">
     <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soap:mustUnderstand="1">{$LOCATION}</wsa:To>
   </soap:Header>
   <soap:Body>
    <csd:careServicesRequest urn="urn:ihe:iti:csd:2014:stored-function:facility-search">
     <csd:requestParams>
      <csd:id/>
      <csd:otherID/>
      <csd:primaryName/>
      <csd:name/>
      <csd:organizations/>
      <csd:codedType/>
      <csd:address/>
      <csd:start/>
      <csd:max/>
      <csd:record/>
     </csd:requestParams>
    </csd:careServicesRequest>
   </soap:Body>
  </soap:Envelope>
 </xforms:instance>

 <xforms:instance id='http-post-request'>
  <csd:requestParams>
   <csd:id/>
   <csd:otherID/>
   <csd:primaryName/>
   <csd:name/>
```

```
3350        <csd:organizations/>
            <csd:codedType/>
            <csd:address/>
            <csd:start/>
            <csd:max/>
            <csd:record/>
3355      </csd:requestParams>
        </xforms:instance>

        <xforms:instance id="soap-response" >
         <soap:Envelope>
3360       <soap:Header>
            <wsa:Action soap:mustUnderstand="1">
             urn:ihe:iti:csd:2013:GetCareServicesResponse
            </wsa:Action>
            <wsa:MessageID/>
3365        <wsa:ReplyTo soap:mustUnderstand="1">
             <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
            </wsa:ReplyTo>
            <wsa:RelatesTo/>
           </soap:Header>
3370       <soap:Body>
            <csd:careServicesResponse>
             <csd:result/>
            </csd:careServicesResponse>
           </soap:Body>
3375      </soap:Envelope>
        </xforms:instance>

        <xforms:instance id="http-post-response" >
         <http:result status="">
3380       <http:header name="X-CSD-Transaction-ID" value=""/>
           <http:body/>
          </http:result>
        </xforms:instance>

3385

        <xforms:submission id="soap-submission" method="post" mediatype="application/soap+xml"
                  ref="instance('soap-request')" resource="{$LOCATION}"/>

        <xforms:submission id="http-post-submission" method="post" mediatype="text/xml"
3390            ref="instance('http-post-request')"
                resource="{$LOCATION}/urn:ihe:iti:csd:2014:stored-function:facility-search" />


        <xforms:action ev:event="xforms-submit-done" id="http-post-prepare">
3395      <xforms:insert
            context="instance('http-post-response')/http:result/http:body"
            origin="instance('care-services-result')/csd:careServicesResponse/csd:result/*"/>
          <xforms:setValue
            bind="instance('http-post-response')/http:result/@status"
3400        value="instance('care-services-result')/csd:careServicesResponse/@code"/>
          <xforms:setValue
            bind="instance('http-post-response')/http:body/@content-type"
            value="instance('care-services-result')/csd:careServicesResponse/@content-type"/>
          <xforms:setValue
3405        bind="instance('http-post-response')/http:result/http:header[@name=
    'X-CSD-Transaction-ID']/@value"
            value="instance('care-services-result')/csd:careServicesResponse/@id"/>
        </xforms:action>

3410    <xforms:action ev:event="xforms-submit-done" id="soap-prepare">
          <xforms:insert
            context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
    @content-type"
            origin="instance('care-services-result')/csd:careServicesRepsonse/@content-type"/>
```

```
3415        <xforms:insert
              context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
            @code"
              origin="instance('care-services-result')/csd:careServicesRepsonse/@code"/>
            <xforms:setValue
3420          bind="instance('soap-response')soap:Envelope/soap:Header/wsa:messageID"
              value="instance('soap-request')csd:careServicesRequest/@id"/>
            <xforms:setValue
              bind="instance('soap-response')soap:Envelope/soap:Header/wsa:relatesTo"
              value="instance('soap-request')soap:Envelope/soap:Header/wsa:messageID"/>
3425        </xforms:action>


            <xforms:bind
              nodeset="instance('soap-request')/soap:Envelope:soap:Header/soap:Body/
3430        csd:careServicesRequest/csd:requestParams/csd:id"
              type="csd:uniqueID" />
            <xforms:bind
              nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
            csd:careServicesRequest/csd:requestParams/csd:otherID"
3435          type="csd:otherID" />
            <xforms:bind
              nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
            csd:careServicesRequest/csd:requestParams/csd:primaryName"
              type="xs:string" />
3440        <xforms:bind
              nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
            csd:careServicesRequest/csd:requestParams/csd:name"
              type="xs:string" />
            <xforms:bind
3445          nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
            csd:careServicesRequest/csd:requestParams/csd:codedType"
              type="csd:codedtype" />
            <xforms:bind
              nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
3450        csd:careServicesRequest/csd:requestParams/csd:address"
              type="csd:address" />
            <xforms:bind
              nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
            csd:careServicesRequest/csd:requestParams/csd:organizations/csd:organization"
3455          type="csd:organization" />
            <xforms:bind
              nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
            csd:careServicesRequest/csd:requestParams/csd:start"
              type="xs:int" />
3460        <xforms:bind
              nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
            csd:careServicesRequest/csd:requestParams/csd:max"
              type="xs:int" />
            <xforms:bind
3465          nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
            csd:careServicesRequest/csd:requestParams/csd:record"
              type="csd:record" />


3470        <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:id"
              type="csd:uniqueID" />
            <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:otherID"
              type="csd:otherID" />
            <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:primaryName"
3475          type="xs:string" />
            <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:name"
              type="xs:string" />
            <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:codedType"
              type="csd:codedtype" />
3480        <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:address"
```

```
        type="csd:address" />
    <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:organizations/
csd:organization"
        type="csd:organization" />
    <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:start"
        type="xs:int" />
    <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:max"
        type="xs:int" />
    <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:record"
        type="csd:record" />

    <xforms:bind nodeset="instance('http-post-response')/csd:CSD" type="csd:CSD" />
    <xforms:bind nodeset="instance('soap-response')/soap:Envelope/soap:Body/
csd:careServicesResponse/csd:result/csd:CSD"
        type="csd:CSD" />

 </xforms:model>
</csd:careServicesFunction>
```

Figure Y.3-3: Facility Search Care Services Function

This is the XML definition of the `careServicesRequest` for Organization Search. A non-normative implementation of the referenced service_search.xq is included in ITI TF-2x: Appendix W.

```
<?xml version="1.0" encoding="UTF-8"?>
<csd:careServicesFunction
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:csd="urn:ihe:iti:csd:2013"
 xmlns:xforms="http://www.w3.org/2002/xforms"
 xmlns:xi="http://www.w3.org/2001/XInclude"
 xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:wsa="http://www.w3.org/2005/08/addressing"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:http="http://expath.org/ns/http-client"
 xmlns:ev="http://www.w3.org/2001/xml-events"
 urn="urn:ihe:iti:csd:2014:stored-function:service-search">
 <csd:description>
   Performs a search for all service by  type,  or ID.
 </csd:description>
 <csd:definition ><xi:include parse='text' href='service_search.xq'/></csd:definition>

 <xforms:model id='urn:ihe:iti:csd:2014:stored-function:service-search'>
  <xforms:instance id="care-services-result">
   <csd:careSerivcesResponse id="" code="" content-type=""/>
  </xforms:instance>

  <xforms:instance id='soap-request'>
   <soap:Envelope >
    <soap:Header>
     <wsa:Action soap:mustUnderstand="1">
       urn:ihe:iti:csd:2013:GetCareServicesRequest
     </wsa:Action>
     <wsa:MessageID/>
     <wsa:ReplyTo soap:mustUnderstand="1">
      <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
     </wsa:ReplyTo>
     <wsa:To soap:mustUnderstand="1">{$LOCATION}</wsa:To>
    </soap:Header>
    <soap:Body>
     <csd:careServicesRequest urn="urn:ihe:iti:csd:2014:stored-function:service-search">
      <csd:requestParams>
```

```
              <csd:id/>
              <csd:otherID/>
3545          <csd:codedType/>
              <csd:start/>
              <csd:max/>
              <csd:record/>
            </csd:requestParams>
3550      </csd:careServicesRequest>
        </soap:Body>
      </soap:Envelope>
    </xforms:instance>

3555  <xforms:instance id='http-post-request'>
       <csd:requestParams>
        <csd:id/>
        <csd:otherID/>
        <csd:parent/>
3560    <csd:start/>
        <csd:max/>
        <csd:record/>
       </csd:requestParams>
      </xforms:instance>

3565
      <xforms:instance id="soap-response" >
       <soap:Envelope>
        <soap:Header>
         <wsa:Action soap:mustUnderstand="1">
3570       urn:ihe:iti:csd:2013:GetCareServicesResponse
         </wsa:Action>
         <wsa:MessageID/>
         <wsa:ReplyTo soap:mustUnderstand="1">
          <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
3575     </wsa:ReplyTo>
         <wsa:RelatesTo/>
        </soap:Header>
        <soap:Body>
         <csd:careServicesResponse>
3580      <csd:result/>
         </csd:careServicesResponse>
        </soap:Body>
       </soap:Envelope>
      </xforms:instance>
3585
      <xforms:instance id="http-post-response" >
       <http:result status="">
        <http:header name="X-CSD-Transaction-ID" value=""/>
        <http:body/>
3590   </http:result>
      </xforms:instance>


      <xforms:submission id="soap-submission"
3595    method="post" mediatype="application/soap+xml"
        ref="instance('soap-request')" resource="{$LOCATION}"/>

      <xforms:submission id="http-post-submission"
        method="post" mediatype="text/xml"
3600    ref="instance('http-post-request')"
        resource="{$LOCATION}/urn:ihe:iti:csd:2014:stored-function:service-search" />

      <xforms:action ev:event="xforms-submit-done" id="http-post-prepare">
       <xforms:insert
3605    context="instance('http-post-response')/http:result/http:body"
        origin="instance('care-services-result')/csd:careServicesResponse/csd:result/*"/>
       <xforms:setValue
        bind="instance('http-post-response')/http:result/@status"
```

```
          value="instance('care-services-result')/csd:careServicesResponse/@code"/>
        <xforms:setValue
          bind="instance('http-post-response')/http:body/@content-type"
          value="instance('care-services-result')/csd:careServicesResponse/@content-type"/>
        <xforms:setValue
          bind="instance('http-post-response')/http:result/http:header[@name=
'X-CSD-Transaction-ID']/@value"
          value="instance('care-services-result')/csd:careServicesResponse/@id"/>
      </xforms:action>

      <xforms:action ev:event="xforms-submit-done" id="soap-prepare">
        <xforms:insert
          context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
@content-type"
          origin="instance('care-services-result')/csd:careServicesRepsonse/@content-type"/>
        <xforms:insert
          context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
@code"
          origin="instance('care-services-result')/csd:careServicesRepsonse/@code"/>
        <xforms:setValue
          bind="instance('soap-response')soap:Envelope/soap:Header/wsa:messageID"
          value="instance('soap-request')csd:careServicesRequest/@id"/>
        <xforms:setValue
          bind="instance('soap-response')soap:Envelope/soap:Header/wsa:relatesTo"
          value="instance('soap-request')soap:Envelope/soap:Header/wsa:messageID"/>
      </xforms:action>


      <xforms:bind
        nodeset="instance('soap-request')/soap:Envelope:soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:id"
          type="csd:uniqueID" />
      <xforms:bind
        nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:otherID"
        type="csd:otherID" />
      <xforms:bind
        nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:start"
        type="xs:int" />
      <xforms:bind
        nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:max"
        type="xs:int" />
      <xforms:bind
        nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:record"
        type="csd:record" />


      <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:id"
        type="csd:uniqueID" />
      <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:otherID"
        type="csd:otherID" />
      <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:primaryName"
        type="xs:string" />
      <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:start"
        type="xs:int" />
      <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:max"
        type="xs:int" />
      <xforms:bind nodeset="instance('http-post-request')/csd:requestParams/csd:record"
        type="csd:record" />

      <xforms:bind nodeset="instance('http-post-response')/csd:CSD" type="csd:CSD" />

      <xforms:bind
```

3675
```
      nodeset="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
csd:result/csd:CSD"
      type="csd:CSD" />

  </xforms:model>
```
3680
```
</csd:careServicesFunction>
```

Figure Y.3-4: Service Search Care Services Function

This is the XML definition of the `careServicesRequest` for Organization Search. A non-normative implementation of the referenced adhoc_search.xq is included in ITI TF-2x: Appendix

3685 W.
```
<?xml version="1.0" encoding="UTF-8"?>
<csd:careServicesFunction
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:csd="urn:ihe:iti:csd:2013"
```
3690
```
 xmlns:xforms="http://www.w3.org/2002/xforms"
 xmlns:xi="http://www.w3.org/2001/XInclude"
 xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:wsa="http://www.w3.org/2005/08/addressing"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```
3695
```
 xmlns:http="http://expath.org/ns/http-client"
 xmlns:ev="http://www.w3.org/2001/xml-events"
 urn="urn:ihe:iti:csd:2014:adhoc"
 >
```
3700
```
<csd:description>
  Performs an adhoc search
</csd:description>
<csd:definition><xi:include href="adhoc_search.xq" parse='text'/></csd:definition>
```
3705
```
<xforms:model id='urn:ihe:iti:csd:2014:adhoc'>
  <xforms:instance id="care-services-result">
   <csd:careSerivcesResponse id="" code="" content-type=""/>
  </xforms:instance>
```
3710
```
  <xforms:instance id='soap-request'>
   <soap:Envelope >
    <soap:Header>
     <wsa:Action soap:mustUnderstand="1">
      urn:ihe:iti:csd:2013:GetCareServicesRequest
```
3715
```
     </wsa:Action>
     <wsa:MessageID/>
     <wsa:ReplyTo soap:mustUnderstand="1">
      <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
     </wsa:ReplyTo>
```
3720
```
     <wsa:To soap:mustUnderstand="1">{$LOCATION}</wsa:To>
    </soap:Header>
    <soap:Body>
     <csd:careServicesRequest  urn="urn:ihe:iti:csd:2014:adhoc">
      <csd:adhoc/>
```
3725
```
     </csd:careServicesRequest>
    </soap:Body>
   </soap:Envelope>
  </xforms:instance>
```
3730
```
  <xforms:instance id='http-post-request'>
   <csd:adhoc/>
  </xforms:instance>
```
```
  <xforms:instance id="soap-response" >
```
3735
```
   <soap:Envelope>
```

```
        <soap:Header>
         <wsa:Action soap:mustUnderstand="1">
          urn:ihe:iti:csd:2013:GetCareServicesResponse
         </wsa:Action>
3740     <wsa:MessageID/>
         <wsa:ReplyTo soap:mustUnderstand="1">
          <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
         </wsa:ReplyTo>
         <wsa:RelatesTo/>
3745    </soap:Header>
        <soap:Body>
         <csd:careServicesResponse>
          <csd:result/>
         </csd:careServicesResponse>
3750    </soap:Body>
       </soap:Envelope>
      </xforms:instance>

      <xforms:instance id="http-post-response" >
3755   <http:result status="">
        <http:header name="X-CSD-Transaction-ID" value=""/>
        <http:body/>
       </http:result>
      </xforms:instance>
3760

      <xforms:submission id="soap-submission"
         method="post" mediatype="application/soap+xml"
         ref="instance('soap-request')" resource="{$LOCATION}"/>
3765

      <xforms:submission id="http-post-submission"
         method="post" mediatype="text/xml"
         ref="instance('http-post-request')"
         resource="{$LOCATION}/urn:ihe:iti:csd:2014:adhoc"/>
3770

      <xforms:action ev:event="xforms-submit-done" id="http-post-prepare">
       <xforms:insert
         context="instance('http-post-response')/http:result/http:body"
         origin="instance('care-services-result')/csd:careServicesResponse/csd:result/*"/>
3775   <xforms:setValue
         bind="instance('http-post-response')/http:result/@status"
         value="instance('care-services-result')/csd:careServicesResponse/@code"/>
       <xforms:setValue
         bind="instance('http-post-response')/http:body/@content-type"
3780     value="instance('care-services-result')/csd:careServicesResponse/@content-type"/>
       <xforms:setValue
         bind="instance('http-post-response')/http:result/http:header[@name=
'X-CSD-Transaction-ID']/@value"
         value="instance('care-services-result')/csd:careServicesResponse/@id"/>
3785   </xforms:action>

      <xforms:action ev:event="xforms-submit-done" id="soap-prepare">
       <xforms:insert
         context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
3790 @content-type"
         origin="instance('care-services-result')/csd:careServicesRepsonse/@content-type"/>
       <xforms:insert
         context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
@code"
3795     origin="instance('care-services-result')/csd:careServicesRepsonse/@code"/>
       <xforms:setValue
         bind="instance('soap-response')soap:Envelope/soap:Header/wsa:messageID"
         value="instance('soap-request')csd:careServicesRequest/@id"/>
       <xforms:setValue
3800     bind="instance('soap-response')soap:Envelope/soap:Header/wsa:relatesTo"
         value="instance('soap-request')soap:Envelope/soap:Header/wsa:messageID"/>
```

```
    </xforms:action>

    <xforms:bind
3805    nodeset="instance('soap-request')/soap:Envelope:soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:adhoc"
      type="xs:string" />

    <xforms:bind
3810    nodeset="instance('http-post-request')/csd:requestParams/csd:adhoc"
      type="xs:string" />

  </xforms:model>
</csd:careServicesFunction>
```

3815 **Figure Y.3-5: Adhoc Care Services Function**

This is the XML definition of the `csd:careServicesRequest` for Query for Updated Services.
A non-normative implementation of the referenced query_for_updated_services.xq is included in
ITI TF-2x: Appendix W.

```
3820  <?xml version="1.0" encoding="UTF-8"?>
      <csd:careServicesFunction
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:csd="urn:ihe:iti:csd:2013"
       xmlns:xforms="http://www.w3.org/2002/xforms"
3825   xmlns:xi="http://www.w3.org/2001/XInclude"
       xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
       xmlns:wsa="http://www.w3.org/2005/08/addressing"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       xmlns:http="http://expath.org/ns/http-client"
3830   xmlns:ev="http://www.w3.org/2001/xml-events"
       urn="urn:ihe:iti:csd:2016:stored-function:query-for-updated-services"
       >
       <csd:description>
        Performs a query for udpated services
3835   </csd:description>
       <csd:definition>
         <xi:include href="query_for_updated_services.xq" parse='text'/>
       </csd:definition>

3840   <xforms:model id='urn:ihe:iti:csd:2016:stored-function:query-for-updated-services'>

        <xforms:instance id="care-services-result">
         <csd:careSerivcesResponse id="" code="" content-type=""/>
        </xforms:instance>
3845
        <xforms:instance id='soap-request'>
         <soap:Envelope >
          <soap:Header>
           <wsa:Action soap:mustUnderstand="1">
3850         urn:ihe:iti:csd:2013:GetCareServicesRequest
           </wsa:Action>
           <wsa:MessageID/>
           <wsa:ReplyTo soap:mustUnderstand="1">
            <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
3855       </wsa:ReplyTo>
           <wsa:To soap:mustUnderstand="1">{$LOCATION}</wsa:To>
          </soap:Header>
          <soap:Body>
           <csd:careServicesRequest
3860         urn="urn:ihe:iti:csd:2016:stored-function:query-for-updated-services">
            <csd:requestParams>
             <csd:lastModified/>
```

```
            </csd:requestParams>
          </csd:careServicesRequest>
        </soap:Body>
      </soap:Envelope>
    </xforms:instance>

    <xforms:instance id='http-get-request'>
     <csd:requestParams>
      <csd:lastModified/>
     </csd:requestParams>
    </xforms:instance>

    <xforms:instance id="soap-response" >
      <soap:Envelope>
       <soap:Header>
         <wsa:Action soap:mustUnderstand="1">
           urn:ihe:iti:csd:2013:GetCareServicesResponse
         </wsa:Action>
         <wsa:MessageID/>
         <wsa:ReplyTo soap:mustUnderstand="1">
           <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
         </wsa:ReplyTo>
         <wsa:RelatesTo/>
       </soap:Header>
       <soap:Body>
         <csd:careServicesResponse>
           <csd:result/>
         </csd:careServicesResponse>
       </soap:Body>
      </soap:Envelope>
    </xforms:instance>

    <xforms:instance id="http-get-response" >
     <http:result status="">
       <http:header name="X-CSD-Transaction-ID" value=""/>
       <http:body/>
     </http:result>
    </xforms:instance>


    <xforms:submission id="soap-submission"
      method="post" mediatype="application/soap+xml"
      ref="instance('soap-request')" resource="{$LOCATION}"/>

    <xforms:submission id="http-get-submission"
      method="get" mediatype="text/xml"
      ref="instance('http-get-request')"
      resource="{$LOCATION}/urn:ihe:iti:csd:2016:stored-function:query-for-updated-services"/>

    <xforms:action ev:event="xforms-submit-done" id="http-get-prepare">
      <xforms:insert
        context="instance('http-get-response')/http:result/http:body"
        origin="instance('care-services-result')/csd:careServicesResponse/csd:result/*"/>
      <xforms:setValue
        bind="instance('http-get-response')http:result/@status"
        value="instance('care-services-result')/csd:careServicesResponse/@code"/>
      <xforms:setValue
        bind="instance('http-get-response')/http:body/@content-type"
        value="instance('care-services-result')/csd:careServicesResponse/@content-type"/>
      <xforms:setValue
        bind="instance('http-get-response')/http:result/http:header[@name=
'X-CSD-Transaction-ID']/@value"
        value="instance('care-services-result')/csd:careServicesResponse/@id"/>
    </xforms:action>

    <xforms:action ev:event="xforms-submit-done" id="soap-prepare">
```

```
      <xforms:insert
3930      context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
@content-type"
          origin="instance('care-services-result')/csd:careServicesRepsonse/@content-type"/>
      <xforms:insert
          context="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
3935  @code"
          origin="instance('care-services-result')/csd:careServicesRepsonse/@code"/>
      <xforms:setValue
          bind="instance('soap-response')soap:Envelope/soap:Header/wsa:messageID"
          value="instance('soap-request')csd:careServicesRequest/@id"/>
3940  <xforms:setValue
          bind="instance('soap-response')soap:Envelope/soap:Header/wsa:relatesTo"
          value="instance('soap-request')soap:Envelope/soap:Header/wsa:messageID"/>
    </xforms:action>

3945  <xforms:bind
          nodeset="instance('soap-request')/soap:Envelope/soap:Header/soap:Body/
csd:careServicesRequest/csd:requestParams/csd:lastModified"
          type="xs:dateTime" />
      <xforms:bind
3950      nodeset="instance('http-get-request')/csd:requestParams/csd:lastModified"
          type="xs:dateTime" />

      <xforms:bind nodeset="instance('http-get-response')/csd:CSD" type="csd:CSD"/>
      <xforms:bind
3955      nodeset="instance('soap-response')/soap:Envelope/soap:Body/csd:careServicesResponse/
csd:result/csd:CSD"
          type="csd:CSD"/>

    </xforms:model>
3960  </csd:careServicesFunction>
```

Figure Y.3-6: Query for Updated Services Care Services Function

3965 # Volume 3 – Content Modules

Not applicable

Volume 4 – National Extensions

3970   Not applicable