

HILT M2M Pilot Requirements Document, V6.0

HILT Phase III is funded to build an M2M version of the HILT Phase II pilot terminologies mapping service, but also to extend its functionality in the various ways described below. The Phase III pilot will not use the Wordmap software used in Phase II, but will employ a simpler-to-use and work on SQL Server based clone as the core of the M2M approach.

At a general level, the requirement is to build an M2M pilot demonstrator service that will:

- Offer web-services access via the (SOAP-based) SRW protocol, but be designed so that a possible extension offering other protocols (Z39.50, or SRU, for example) at a later date could be an option.
- Use SKOS-Core as the ‘mark-up’ for sending out terminology sets and classification data responses but be designed so that adding other formats such as MARC and Zthes would be an option at a later date.
- Provide the pilot datasets, mappings, and functionality capable of servicing the 5 use cases¹ agreed in the HILT M2M Feasibility Study (see Appendix C).
- Be based on the centralised approach to the provision of mapping services piloted in HILT Phase II, but be designed so that the possibility of a future move towards a more distributed model is kept open.
- Include a pilot implementation of the SRW ‘EXPLAIN’ function.

This document is intended to specify the requirement in more detail. Some of the detail of the requirement may be amended at any time up to the project end date. Future (i.e. post-Phase III) requirements will also be specified here (as Appendix B).

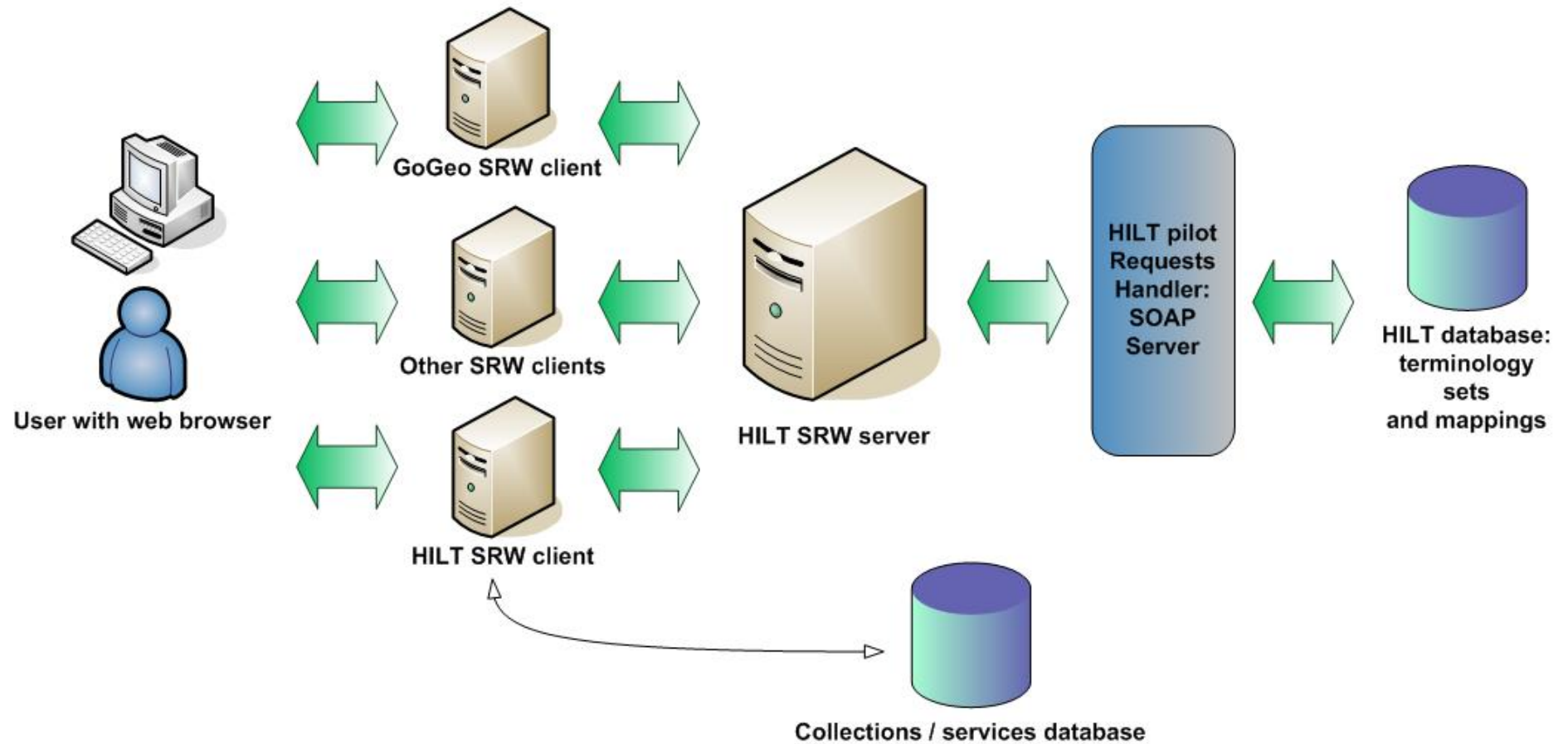
Note that the primary aim is to specify what the functional requirements are, rather than how they will be met.

Version	Date	Comments	Version	Date	Comments
Version 1	25.11.2005	Initial draft - Dennis Nicholson (DN).			
Version 2	07.03.2006	DN (post discussions with Edina).			
Version 3	20.04.2006	DN			
Version 4	28.08.2006	George Macgregor (GM)			Addition of HILT M2M SOAP functions table (Appendix D).
Version 5	20.10.2006	Updated and revised by GM			Updated and standardised SOAP function labels (Appendix D)
Version 6	27.10.2006	Updated (EM)			Addition of DN’s Functions and use cases

¹ Note that the original use cases suggested that Phase III investigate whether certain use case functionality elements would be best provided at the client end or at the HILT end. On this front, it has been determined that it is more in keeping with the architecture to assume that, in the main, functionality will be provided at the client end unless there is good reason to do otherwise.

					document (Appendix E)
--	--	--	--	--	-----------------------

HILT M2M Pilot Architecture



Section 1: Primary Elements and their Functions

Introduction

The diagram on page 2 above shows the primary elements of the M2M pilot, together with a rough representation of their inter-relationships. This section - Section 1 - provides an overview of the functions performed by each element, together with some associated top level requirements and explanatory notes. Detailed functional requirements for the whole (pilot) system are dealt with in Section 2 below (page 7)

Web Browsers

Users will not access HILT directly as they did in the Phase II pilot. They will use web browsers to access services such as GoGeo! and Biome and the services will interact with HILT via service-based SRW clients and a HILT SRW server. In the planned Phase III pilot, however, these clients will essentially be web pages accessed via user browsers. Since this is a pilot implementation only, the current assumption is that these client sites will support only Internet Explorer 6x. Since web browsers are not part of the pilot proper, and this is the only requirement associated with them, they are not included in the tables showing the detailed requirement (Section 2 below).

GoGeo! Service SRW Client

This SRW client will entail two kinds of requirements:

- Generic requirements likely to be incorporated in any client - requirements related to the sending of requests to, and the displaying or using of responses from, the HILT terminologies mapping service in the servicing of the 5 use cases.
- GoGeo! specific requirements – special requirements related to the specific needs of the service and its users.

Its functions are:

- To translate service requests for SRW ‘explain’ information into Common Query Language (CQL) requests and ‘wrap’ them in SOAP-like SRW markup for sending to HILT via the SRW server.
- To translate either direct user requests for terminologies information or, more often, service mediated requests that ‘interpret’ user needs, into Common Query Language (CQL) requests and ‘wrap’ them in SOAP-like SRW markup for sending to HILT via the SRW server.
- To deal with and, as appropriate, display the responses to these requests.

The GoGeo! Service SRW Client will be written in the [as yet unspecified] programming language.

HILT Service SRW Client

In the first instance, this SRW client will emulate the current user interface to HILT but extend its functionality to cover the detail of all 5 use cases. However, it may also be extended to explore additional service specific requirements of a few representative services in UK HE and/or FE.

Its functions are:

- To translate service requests for SRW ‘explain’ information into Common Query Language (CQL) requests and ‘wrap’ them in SOAP-like SRW markup for sending to HILT via the SRW server.
- To translate either direct user requests for terminologies information or, in the case of service specific functions, service mediated requests that ‘interpret’ user needs, into Common Query Language (CQL) requests and ‘wrap’ them in SOAP-like SRW markup for sending to HILT via the SRW server.
- To deal similarly with queries to the collections database for collections information
- To deal with and, as appropriate, display the responses to these requests.

The current assumption is that the HILT Service SRW Client will be written in the Java programming language. Ideally, it could be the basis of an open source baseline client that other JISC IE services could take and adapt to their own needs (note, however, that this is not a deliverable).

SRW Server

This will:

- Take the SRW-wrapped CQL requests coming from the clients, unwrap them, determine whether they are Explain requests or terminology requests, translate the CQL commands and syntax into commands and syntax recognised by the SRW Explain function, or the HILT requests and responses handler, and send them on to the service function in question.
- Take responses from these various services, translate them into CQL responses and syntax, wrap them in SRW markup, and send them back to the client with labels to indicate their source. In the case of responses from the main HILT database, which will be wrapped in SKOS-Core, the SRW markup will encompass the SKOS-Core markup rather than replace it.

If it is allowed by the SRW Explain function, there may be a case for the SRW server being able to download data on the HILT command and response language via the SRW Explain file. Failing that, there is probably a future case for it being able to identify (say) all of the terminology servers in a distributed HILT (or, indeed, elsewhere) in a service like IESR and download information on their command and response language

from there. Probably, though, the safe approach as regards reliability of the data, is for the SRW server to be able to identify terminology services through IESR and command and response language through SRW Explain.

Requests and Responses Handler

This is where the functionality relating to querying the HILT terminologies database will reside. It will:

- Take requests from the SRW server, map these to appropriate APIs designed to optimise searching/retrieving from the various terminology sets, mappings, and other data, and, via these, send search requests to the HILT terminologies database.
- Take responses from the database, wrap them in SKOS-Core mark-up, process them as (and if) required, and send the responses back to the SRW server for passing on to the client(s). In some cases – possibly in all – it would also pass on information regarding the subject scheme that was the source of a term or terms sent back. In a future implementation, it should be possible for it to allow the client to specify which mark-up is required from a range available via HILT (e.g. Zthes) and use the mark-up chosen. In this event, the range available will presumably be specified in the SRW Explain file. The details of the mark-up themselves might be available from the HILT database (we should assume so in the pilot and think about putting SKOS-Core details in the database), although it could be sensible for a range of standard mark-ups to be available in some kind of separate registry.
- In some cases, react to a ‘no hits from HILT’ response from the database by offering a browse of DDC or of an individual scheme and/or a google-type ‘did you mean...’ this response (i.e. was that a spelling mistake?).

In many cases a given service-mediated user query may entail a number of connected interactions with the database, but the current assumption is that it will be possible to handle this situation without the requests and responses handler having to keep track of the ‘thread’ of an individual query. Ideally, the design of the overall system would be such that any keeping track required could be handled at the client end and the requests and responses handler need only handle apparently unrelated requests. Presumably, though, it would have to keep information regarding the source of the original query and some kind of query number assigned by the client/querying service. This would mean it could keep track of a query if necessary, and could, for example, produce query-related statistics should this be required at some future date. At the moment, however, the assumption is that, functionality-wise, keeping track of a query is a client end function.

In the pilot at least, it will probably make sense for it to also handle requests to and responses from the Explain service. Dealing with the Explain service in this way may still make sense in an operational situation in the longer term (not sure).

HILT Terminologies Database

This is where the various terminologies sets, mappings and other data available through HILT are held. The current assumption is that, in the pilot, it will hold DDC, OCLC provided mappings from LCSH to DDC, MeSH, together with some illustrative mappings to DDC, UNESCO,

together with some illustrative mappings to DDC, JACS, together with some illustrative mappings to DDC, IPSV, together with some illustrative mappings to DDC, a sample of AAT, together with some illustrative mappings to DDC.

In order to permit a possible future move to a distributed terminologies and inter-scheme mapping service, it will be possible to assign an SKOS-Core concept URIs to terms in the schemes and an illustrative range of these would be held for each of the schemes noted above.

The database will also hold information on relationships between terms in schemes and, for mappings between schemes, information on the type of mapping (exact match, singular-plural etc).

Illustrative information on common spelling errors will also be held.

Collections Description Service

This is intended to be a stand-in for IESR. In the pilot, it should be based on the version used in HILT Phase II but should use the multiple DDC assignments now used in IESR to give a better identification service. Where appropriate, IESR now lists more than once DDC number for some services. Ideally, this would also be SRW-compliant, but it is not essential for the pilot that it should be.

SRW Explain

This should hold data on the subject schemes used, the kinds of queries supported, the kinds of terminologies mark-ups offered (SKOS-Core only at present), and so on. More research is needed into what is permitted by the protocol and what is needed for the service in both a centralised, standalone mode, and a distributed one, before full details of the requirement can be specified.

General Functionality and Variant Cases

The pilot will be designed to permit an exploration of the circumstances described under ‘Variant Cases’ in the use cases description – for example, the effects of using phrases rather than single words as a search term, or the effects of terms with large mappings.

Section 2: Detailed Functional Requirements

Pilot Functionality Scope

The functionality proposed for the pilot comes from four sources:

1. The functionality of the (non-M2M) Phase II pilot, as amended by discussions within the HILT M2M Feasibility Study. This is Use Case #1 in Appendix C below.
2. The remaining use cases (Use Case #2 to Use Case #5) specified in Appendix C below.
3. Additional functionality suggested by ongoing work within the current phase of HILT (Phase III) and accepted as being realisable within project resources.
4. The generalisation of facilities required in one use case, or for one scheme, to other use cases or schemes.

Pilot Functionality and Use Cases

Since the various use cases implicitly or explicitly entailed overlapping functional requirements, the functional requirements (listed below as Appendix A (page 8)) have not, in the main, been subdivided by use case. The exception is client functionality over and above simple request, receive, and present. This division into simple/generic and advanced client functionality will permit development to be carried out on information exchange prior to tackling more complex questions related to the handling of the information retrieved. As the pilot is developed, new requirements may be added, and some of the detail of how current requirements are expressed, may change as unpredicted issues are raised and resolved.

Other Functionality

Additional functional requirements identified within Phase III, but not realisable within its resources, will be listed in Appendix B.

General Note on Functionality Requirements

In simple terms, the functionality requirements fall under six headings:

- Interaction start-up functions (mainly SRW explain related), and interaction close down functions, as required.
- Database coverage and function.
- Collections database coverage and record content.
- Simple, whole-system request, transmit/receive, and present functions. These look complex when described in detail, but boil down to a requirement to be able to do one of three things: **(1)** request, transmit/receive, and present everything in the database on some **unqualified** term or term-phrase specified by a user via a service client (everything required at the client end to provide all of the functionality available

in the Phase II pilot – which is to say, everything required by use case #1 with the stage 2 switch **on** (see Appendix C)), (2) request, transmit/receive, and present everything required by use case #1 but carried out in two steps instead of one – i.e. with the stage 2 switch **off** (essentially, get all of the DDC mappings associated with a term, then, when user chooses one of them, get every record in the database that DDC number is mapped to, (3) request, transmit/receive, and present everything in the database on a term or term-phrase **qualified** in one or more ways so as to limit data returned (mappings only, say) or subject schemes to be included in a query (LCSH and DDC only, say). Note that there is a good deal of functionality buried in (3).

- Use case specific client functionality, which is more complicated, and difficult to specify properly without more research (except, perhaps, for Use Case #1). A problem is that the use cases are really very vague when closely scrutinised. As a consequence, it will be necessary to examine and specify what has to be done at the client end to implement the use cases in the light of a more detailed examination of the data to be handled and the various screen design and functionality issues the use cases throw up when faced with real world situations and real terminological data.
- Error-handling functionality

Appendix A: Pilot Functionality Requirements.

Function	Note
Start-up, explain, close-down functionality	
1. Initiate / close down information exchange session.	Are these necessary? Is close-down unnecessary because client-server? Is initiate implicit in any information request client sends?
2. Store an SRW explain file with details of functionality, schemes, mappings, other data held, available mark-ups, and so on.	
3. Request SRW explain file	Either client must be able to request; remainder of the system (other than the collections database) must be able to recognise and process the request.
4. Transmit SRW explain file	HILT requests and responses handler should be able to retrieve the file from the database and send it to the requesting client; other parts of the system should be able to process the response appropriately so that it reaches the requesting client.
5. Present SRW explain file	Client must be able to display the Explain file
6. Process SRW explain file	Client must be able to process the Explain file – that is, extract details of functionality etc in a form that can then be used by the client to meet the various terminology-related needs of its users.

<p>7. Use the information held in the SRW explain file</p>	<p>Client must be able to use the information held in the SRW explain file to meet the various terminology-related needs of its users.</p>
<p>Database coverage and functionality</p>	
<p>8. Store a range of subject and/or classification schemes and related data, sometimes completely, sometimes selectively. Data in each record to include (where they exist) at least term, scope note, narrower and broader terms, synonyms, related terms, singular/plural information, mappings to DDC, SKOS Core Concept URI, (this list under construction). Individual elements to be individually addressable.</p>	<p>Schemes currently being examined: DDC, LCSH (mappings to DDC only), MeSH, UNESCO, IPSV, JACS. Selected terms within schemes will hold mappings to DDC, match types, and SKOS Core concept URIs.</p>
<p>Collections database coverage and record content</p>	
<p>9. Use same collections database design as in HILT II but extend some collections records to hold addition DDC numbers in order to better describe the subject content of collections and better reflect the data held in the new IESR.</p>	
<p>Whole-system request, transmit/receive, and present functions</p>	
<p>10. Ability to request, transmit/receive, and present everything in the database on some unqualified term or term-phrase specified by a user via a service client. Specifically, the system should, in response to a term or term phrase that is not qualified in any way, send back and present:</p> <ul style="list-style-type: none"> ○ Everything in every record in the database (including DDC records) that has the specified term in it and ○ Everything in every DDC record that any non-DDC record in the above set is mapped to, but excluding (i.e. not duplicating) any DDC record already in the above set and ○ Everything in every record mapped to any of the DDC records retrieved, but excluding (i.e. not duplicating) any non-DDC record already in the retrieved set. <p>Essentially, the need is to retrieve – without duplicates - everything in any record in the database that either contains the term, is a DDC record mapped to from a record that contains the term, or is a non-DDC record mapped to from a DDC</p>	<p>This is all of the information on a term or term-phrase required at the client end to provide all of the functionality available in the Phase II pilot – which is to say, everything required by use case #1 with the stage 2 switch on (see Appendix C). It is not, however, all of the functionality required. Only the request and present client functionality is covered. The remainder of the required functionality is covered under the ‘use case specific functionality’ heading below (see under use case #1).</p> <p>Client steps beyond simple present for which the information retrieved is required:</p> <ul style="list-style-type: none"> ○ The client takes a user term, presents the user with any DDC record that either contains it or is mapped to by a non-DDC record that contains it and asks the user to choose one ○ Once a DDC record (and, hence, number) is chosen the client uses it to search the collections database using successive truncations of the DDC number. It retrieves data on collections relevant to a user’s subject term, the subject

<p>record that either contains the term or is mapped to from a non-DDC record that does. The last group ensures that all mappings from a relevant DDC term to other schemes known about are encompassed.</p> <p>Note that it will be essential to ensure that the mark-up makes clear which scheme a record belongs to. Note also that, whilst it may not be necessary to retrieve everything in every retrieved record, it may be simpler to do so, and the aim of covering this function in one go is to make it simpler.</p>	<p>scheme it uses.</p> <ul style="list-style-type: none"> ○ It then uses the non-DDC records mapped to the DDC number chosen by the user to identify the appropriate term in a given subject scheme to use when searching a collections using that scheme ○ Where possible, the collection concerned is searched using that term.
<p>11. Ability to request, transmit/receive, and present everything in every DDC record in the database that either contains some unqualified term or term-phrase specified by a user via a service client or is mapped to by a non-DDC record that contains it.</p>	<p>The functionality described in item 10 above can be provided in two steps, of which this is step 1. Only the DDC records from the full set described in the row above are sent and presented.</p> <p>This is all of the information on a term or term-phrase required at the client end to provide the first part of the functionality available in the Phase II pilot – which is to say, everything required by use case #1 with the stage 2 switch of (see Appendix C). It is not, however, all of the functionality required. Only the request and present client functionality is covered. The remainder of the required functionality is covered under the ‘use case specific functionality’ heading below (see under use case #1).</p> <p>Client steps beyond simple present for which the information retrieved is required:</p> <ul style="list-style-type: none"> ○ The client takes a user term, presents the user with any DDC record that either contains it or is mapped to by a non-DDC record that contains it and asks the user to choose one ○ Once a DDC record (and, hence, number) is chosen the client uses it to search the collections database using successive truncations of the DDC number. It retrieves data on collections relevant to a user’s subject term, the subject scheme it uses.

<p>12. Ability to request, transmit/receive, and present everything in every record in the database mapped to a particular DDC number</p>	<p>The functionality described under item 10 above can be provided in two steps, of which this is step 2. Only the non-DDC records mapped to the DDC number sent are retrieved.</p> <p>This is all of the information on a term or term-phrase required at the client end to provide the second part of the functionality available in the Phase II pilot – which is to say, everything required by use case #1 with the stage 2 switch of (see Appendix C). It is not, however, all of the functionality required. Only the request and present client functionality is covered. The remainder of the required functionality is covered under the ‘use case specific functionality’ heading below (see under use case #1).</p> <p>Client steps beyond present for which the information retrieved is required:</p> <ul style="list-style-type: none"> ○ The non-DDC records mapped to the DDC number chosen by the user is used to identify the appropriate term in a given subject scheme to use when searching a collection using that scheme ○ Where possible, the collection concerned is searched using that term.
<p>13. Ability to request, transmit/receive, and present everything in the database on some qualified term or term-phrase specified by a user via a service client – that is, to limit a search to specified subject schemes (LCSH and UNESCO only, say), or to limit retrieval to specific data elements (mappings only, related terms or narrower terms or scope notes only), or to do a combination of these things.</p>	<p>Needed for use cases #2 to #5 and of possible application in creating improved functionality in use case #1</p>
<p>Use-case specific client functionality</p>	
<p>14. Ability to emulate HILT Phase II pilot functionality in HILT SRW client (only) using both the one stage process described at 10 above and the two stage process described at 11 and 12 above and the different retrieval sets entailed in each of these processes. If feasible, deal with the situation where requesting</p>	<p>See notes above for details</p>

<p>services, or services identified through collections database, use more than one subject scheme. Deal also with situation where collections database assigns more than one subject term to collections it describes.</p>	
<p>15. Functionality to match the following outline intelligently:</p> <ul style="list-style-type: none"> ○ User types a term into service-end search box. Term is sent to HILT to generate an additional set of search terms that can be queried against the sending service database. ○ Web form created listing the original term, and the initially expanded/ derived terms, and presented back to the user ○ User given feedback on origin of derived term. ○ User selects terms from Web form for further expansion via HILT The results of the expansion are then inserted into the Web form. ○ User gets functions to: <ul style="list-style-type: none"> ➤ Map plural to singular terms; Map synonyms to main terms in thesauri; disambiguate terms such as COLD; Correct simple spelling/typographic errors. ➤ Having used these various functions, user selects one or more terms derived from the mapping process and these are used to search the requesting service database. Results are displayed in browser without substantial differences to the non-enhanced search. 	<p>Further R&D required before full use case #2 requirement can be specified.</p>
<p>16. Functionality to match the following outline intelligently:</p> <ul style="list-style-type: none"> ○ User types a term into search box. The term is sent to HILT to generate a set of additional search terms that can be used to search the requesting service database. ○ If any simple spelling or typographical errors are identified an intermediate screen offering an alternative spelling is presented along the lines of Google, "Did you mean?" ○ After acquiring a correct spelling the term is sent back to HILT for further expansion. ○ The original and derived terms are passed to the requesting service database, a search is run against it and a result set is 	<p>Further R&D required before full use case #3 requirement can be specified.</p>

returned. The user notices no substantial differences in the result set (apart from hopefully a larger number of results) between the non-enhanced query and a query enhanced first by via M2M interaction with HILT.	
17. Functionality to match the following outline intelligently: <ul style="list-style-type: none"> ○ Browse (of DDC) offered by client in response to a ‘no hits from HILT’ situation in response to a service-end request. ○ Browse of appropriate scheme offered by client when requested by user in response to a particular term provided by HILT from the scheme in question. 	Further R&D required before full use case #4 requirement can be specified.
18. Functionality to match the following outline intelligently: <ul style="list-style-type: none"> ○ Improved precision based use cases, for example: client request for information on narrower and related terms and (possibly) cross-scheme variations on this. 	Further R&D required before full use case #5 requirement can be specified.
Error handling functionality	
19. Deal with any unrecognised error by transmitting a ‘request not recognised’ error message, together with the Explain file	
20. Where a term is not recognised. Assume a spelling or typographical error has been made and send the client close matches with a Google, "Did you mean?" message (although, probably, in this case there would often be more than one option	Is this feasible in a pilot where a term may not be recognised even if it is spelt correctly? Better to associate common mis-spellings and typos with individual terms? Perhaps this would be functionally better (if somewhat labour intensive) even in an operational situation?
21. In response to a ‘no hits’ result in a database search do as specified under Use Case #4, part 1	
22. If necessary, deal with response sets so large that they cause performance problems (e.g. if use case #1 switch is set on)	

Appendix C: Use Cases.

[**Note that** the use cases below have been copied verbatim from the HILT M2M feasibility Study documentation and may not, in the last analysis, be implemented entirely as specified if changes are indicated by ongoing Phase III work and analysis]

Use case #1

Single two-stage process with a ‘switch’ used to turn stage two on and off.

Client sends request to HILT server for data on a subject search term (‘teeth’, say).

If request stage two switch at **off**, and teeth is the term, the server applies the Wordmap (or equivalent) search_for_wordsets function with teeth as ‘search_term’ parameter and returns all senses of wordsets (wordset id and the tree) that have word phrases that match ‘teeth’.

If request stage two switch at **on**, server **also** applies the Wordmap (or equivalent) get_features function and returns, **in addition**, a record for each feature of the wordset. The features retrieved are Dewey number associated with a term, and the mappings available. For example, in the case of one possible result of a search for teeth, the Dewey number is 611.314, and mappings are held in the database for LCSH (statistical mapping) and the MeSH taxonomy (singular plural match).

Some services would do the above in a single call, others as two separate calls. The use of the DDC number to search for appropriate collections in IESR would be a service end function, although HILT would also provide that option as an additional call and would maintain the code for the DDC algorithm and make it available to the community. Disambiguation would be a service end function based on data sent back from HILT.

Use case #2

BIOME/Go Geo!/RDN #1.

User types a term into service-end search box. Term is sent to HILT to generate an additional set of search terms that can be queried against the sending service database.

Web form created listing the original term, and the initially expanded/ derived terms, and presented back to the user

User given feedback on origin of derived term.

User selects terms from Web form for further expansion via HILT The results of the expansion are then inserted into the Web form.

~~~

User gets functions to:

Map plural to singular terms; Map synonyms to main terms in thesauri; disambiguate terms such as COLD; Correct simple spelling/typographic errors.

~~~

Having used these various functions, user selects one or more terms derived from the mapping process and these are used to search the requesting service database. Results are displayed in browser without substantial differences to the non-enhanced search.

~~~

The use case should allow for two possibilities – one is that user interaction is all handled at requesting service end rather than HILT end, the other that HILT will handle the interaction. The question of which is the best/most practical/most economic approach is most likely to be examined in the context of the likely M2M demonstrator project.

### Use case #3

BIOME/Go Geo!/RDN #2.

~~~

User types a term into search box. The term is sent to HILT to generate a set of additional search terms that can be used to search the requesting service database.

~~~

If any simple spelling or typographical errors are identified an intermediate screen offering an alternative spelling is presented along the lines of Google, "Did you mean?"

~~~

After acquiring a correct spelling the term is sent back to HILT for further expansion.

~~~

The original and derived terms are passed to the requesting service database, a search is run against it and a result set is returned. The user notices no substantial differences in the result set (apart from hopefully a larger number of results) between the non-enhanced query and a query enhanced first by via M2M interaction with HILT.

~~~

The question of whether it is better/more practical/ more economic for HILT to provide the 'did you mean' interface (as opposed to just the data that drives it) is again one for the future M2M demonstrator project.

Use case #4

Browse-based use cases.

~~~

Four situations to consider have been identified under this heading:

(a) Browse offered by HILT in response to a ‘no hits from HILT’ situation in response to a service-end request.

~~~

(b) Browse of appropriate scheme offered by HILT when requested by user in response to a particular term provided by HILT from the scheme in question.

~~~

(c) Browse of (a) handled by requesting service rather than by HILT.

~~~

(d) Browse of (b) handled by requesting service rather than by HILT.

~~~

### Use case #5

Improved precision based use cases.

~~~

Two of these are covered by browse use cases, and by disambiguation in use-cases 1 and 2.

~~~

We should probably also consider requests to HILT for information on narrower and related terms and (possibly) cross-scheme variations on this.

### Variant Cases

Consideration needs to be given to the effects of having a phrase as the search term and of the effects of terms with large mappings. Are there searches or circumstances for which the effects of having the second stage switched on are such that they hit response times and where result-sets are excessively large? Also, are there cases where services that are running the DDC IESR search need to make additional calls to the HILT server for supplementary information? Use cases also need to consider the situation where requesting services, or services identified through IESR, use more than one subject scheme.

**Appendix D: HILT SOAP Functions Table**

|          | <b>Function name</b> | <b>Description</b>                                                                                                                                                           | <b>Use Case</b>                                                 | <b>Notes</b>                                                                                                                                                                      |
|----------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1</b> | explain              | Get HILT service Explain file (covers requirements 2-7)                                                                                                                      | All                                                             | If at CDLR (not clear)                                                                                                                                                            |
| <b>2</b> | get_collections      | Get collections classified under specified DDC number or its stem (covers requirement 9 - retrieval here is from separate collections database and might ultimately be IESR) | Specific usage in is #1 but might be used in other contexts     | May not be necessary if collections database accessed directly or via (CDLR-based) SRW rather than SOAP.                                                                          |
| <b>3</b> | get_all_records      | Get records that include – or are directly or indirectly mapped to records that include – specified term or term phrase (covers requirement 10)                              | #1, probably not used in other contexts.                        |                                                                                                                                                                                   |
| <b>4</b> | get_DDC_records      | Get any DDC record that either includes the term specified, or that is mapped to by a record that includes the term specified (covers requirement 11).                       | #1, but might be used in other contexts                         |                                                                                                                                                                                   |
| <b>5</b> | get_non_DDC_records  | Get any non-DDC record that includes a mapping to the DDC number sent (covers requirement 12).                                                                               | #1, but might be used in other contexts.                        |                                                                                                                                                                                   |
| <b>6</b> | get_filtered_set     | Get record and fields set that meets the specified parameters (covers requirement 13)                                                                                        | Various different parameter sets needed for use cases #2 to #5. | Filters would be by things such as subject scheme (UNESCO and LCSH only, say), or specified fields (mappings only, say, or broader terms and narrower terms only, say) and so on. |

## **Appendix E: HILT Functions and Use Cases**

### **Use Case 1:**

Emulate the Phase II pilot: use the user's search term to identify DDC captions that might cover his or her topic and get user to 'disambiguate' captions (i.e. identify which does cover his or her topic) (Get\_DDC\_records); use DDC number associated with caption to identify collections that cover the subject area in which the user's topic 'sits' and the subject schemes they use and get user to choose collection to search and, in so doing, identify the subject scheme to use when searching(Get\_collections);; identify term in the scheme in question to use when searching for user's topic; search for it and return hits to user (Get\_non\_DDC\_records).

Also, get\_all\_records combines Get\_DDC\_records and Get\_non\_DDC\_records

### **Use Cases 2/3:**

Provide enriched set of search terms so users can use (selectively or otherwise) to improve searches of local service database; includes term expansion, singular/plural mappings, spelling/typos correction (Google 'did you mean?') – either get\_all\_records or get\_filtered\_set.

### **Use Case 4:**

Scheme hierarchy browse in 'no hits from HILT' or user/service request situation (get\_filtered\_set).

### **Use Case 5:**

Improved precision use cases, either via browsing scheme hierarchies, disambiguation or specific requests for narrower/related terms (other use cases plus get\_filtered\_set).