Goals

- Consider proposal to change language syntax from ABNF to ANTLR
- Recap ECL transitivity/role chain proposal, and consider implementation support
- Summarize 2019 work items, including:
  - Proposed new ECL language features
  - Updates to URI standard
  - Enhancement to template language
  - Draft Query Language

Attendees

- Chair: Linda Bird
- Project Group: Daniel Karlsson, Michael Lawley, Anne Randorff Hejen, Kai Kewley, Ed Cheetham, Rob Hausam, Guillermo Reynoso, Harold Solbrig

Agenda and Meeting Notes

<table>
<thead>
<tr>
<th>Description</th>
<th>Owner</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome and apologies</td>
<td>Linda Bird</td>
<td>For discussion: Daniel has proposed to change the primary syntax representation for all languages from ABNF to ANTLR.</td>
</tr>
</tbody>
</table>
| Proposal to move from ABNF to ANTLR | Daniel Karlsson | Auto-translation from ABNF to ANTLR difficult
ABNF grammar represents the byte representation of UTF-8 / Needs to be updated to include a range of UTF characters (not the byte representation)
ABNF is a tool-independent language for defining syntaxes / ANTLR is a tool for defining parsers (and dependent on versions)
Proposal - Continue to use ABNF as the standard, with a hand-crafted ANTLR as the authoritative ANTLR reference implementation
Alternate representations, e.g. JSON, would be useful (e.g. for templates)
Actions
  - Post authoritative ANTLR syntax used by SNOMED International
  - Update ABNF with additional UTF characters |
| Transitivity, Reflexivity & Role chaining | Linda Bird Kai Kewley | Proposed extension to ECL to support transitive relationships and role chaining (to align with new enhanced DL axioms) |
| | | Example 1 Direct relationship
< [Body structure]: << 774081006 |Proper part of| = << 51185008 |Chest|
Transitive relationship
  - < [Body structure]: << 774081006 |Proper part of| = << 51185008 |Chest|
  - <<| [Body structure]: << 774081006 |Proper part of| = << 51185008 |Chest|
Example 2 Direct relationship
  - < 71388002 ||: 363701004 |Direct substance| = 372687004 |Amoxicillin|
Role chained relationship (via 738774007 [Is modification of])
  - < 71388002 ||: 363701004 |Direct substance|* = 372687004 |Amoxicillin|
  - <<| 71388002 ||: 363701004 |Direct substance| = 372687004 |Amoxicillin|
What implementation support will be required? Should we provide easy access to those relationships that can be inferred by transitivity and role chains (note: These will be excluded from the inferred relationship file as they are redundant). If so, then what format should be used - for example, a TSV file with the following columns:
  - sourceld
  - destinationId
  - typeld
  - relationshipGroup |

END OF MEETING
<table>
<thead>
<tr>
<th>Executing maps</th>
<th>Linda Bird</th>
<th>Proposed extension to ECL to support the execution of maps (focusing on the resolution of historical refsets)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>- The specific use-case here comes initially from Jeremy and relates to being able to work with inactive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>concepts via the historical association maps. For example, given an ECL expression that identifies a set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of concepts 'c' to be used for retrieving patient records, you probably also want to retrieve records for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sameAs (c) and replacedWith (c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>`&lt; 72704001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Template Syntax</th>
<th>Linda Bird</th>
<th>New requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>- 2 slots must have the same value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 slots must have different values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The value of 1 slot must subsume the value of another</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Default value for slots</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URI Standard</th>
<th>Linda Bird</th>
<th>Finalize and publish language and language instance URIs</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Query Language - Summary from previous meetings</th>
<th>Linda Bird</th>
<th>Examples: version and language</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>- `&lt;&lt; 64572001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- `&lt;&lt; 64572001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- `&lt;&lt; 64572001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- `&lt;&lt; 64572001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- `&lt;&lt; 64572001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>(*) ([ term = &quot;heart&quot; ])) VERSION http://snomed.info/sct/9000000000000207008/version/20180131</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>(*) ([ term = &quot;heart&quot; ])) VERSION http://snomed.info/sct/9000000000000207008/version/20170731</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>X MINUS Y WHERE X = *, Y = (*) ([ term = &quot;heart&quot; ])) VERSION http://snomed.info/sct/9000000000000207008/version/20180131</code></td>
</tr>
</tbody>
</table>

**Notes**
- Allow nested where, version, language
- Scope of variables is inner query

<table>
<thead>
<tr>
<th>Examples: where</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X MINUS &gt;! X WHERE X = (&lt; 1234 : 5678 = &lt; 6547)</td>
<td></td>
</tr>
<tr>
<td>X MINUS &gt;! X WHERE X = (&lt; 1234 : 5678 = &lt; 6547) VERSION <a href="http://snomed.info/sct/9000000000000207008/version/20180131">http://snomed.info/sct/9000000000000207008/version/20180131</a></td>
<td></td>
</tr>
<tr>
<td>X MINUS &gt;! Y WHERE X = (&lt; 1234 : 5678 = &lt; 6547), Y = (&lt; 1456) VERSION <a href="http://snomed.info/sct/9000000000000207008/version/20180131">http://snomed.info/sct/9000000000000207008/version/20180131</a></td>
<td></td>
</tr>
<tr>
<td>X MINUS &gt;! X WHERE X = (&lt; 1234 : 5678 = &lt; 6547) VERSION <a href="http://snomed.info/sct/9000000000000207008/version/20180131">http://snomed.info/sct/9000000000000207008/version/20180131</a>, LANGUAGE 999001881000000108</td>
<td>GB clinical extension LRS</td>
</tr>
<tr>
<td>X MINUS &gt;! X WHERE X = (&lt; 1234 : 5678 = &lt; 6547) VERSION <a href="http://snomed.info/sct/9000000000000207008/version/20180131">http://snomed.info/sct/9000000000000207008/version/20180131</a>, LANGUAGE 999001881000000108</td>
<td>GB clinical extension LRS</td>
</tr>
<tr>
<td>X minus &gt;! X WHERE X = ( &lt; M WHERE M = (&lt; 1234)) VERSION <a href="http://snomed.info/sct/9000000000000207008/version/20180131">http://snomed.info/sct/9000000000000207008/version/20180131</a>, LANGUAGE 999001881000000108</td>
<td>GB clinical extension LRS</td>
</tr>
</tbody>
</table>

**Notes**
- Allow nested variable definitions, but recommend that people don't due to readability
- Scope of variables is the inner query
- No recursion e.g. X WHERE X = 1234 MINUS X
  - ie can't use a variable in its own definition
  - ie X is only known on the left of the corresponding WHERE, and not on the right of the WHERE
Keywords for Term-based searching:

- **D.term**
  - D.term = "*heart"
  - D.term = wild:"*heart"
  - D.term = regex:"*heart"*
  - D.term = match:"hear att"
  - D.term = (sv) wild: "*heart"

- **D.languageCode**
  - D.languageCode = "en"
  - D.languageCode = "es"

- **D.caseSignificanceId**
  - D.caseSignificanceId = 900000000000448009 |entire term case insensitive|
  - D.caseSignificanceId = 900000000000170005 |entire term case sensitive|
  - D.caseSignificanceId = 900000000000020002 [only initial character case insensitive]

- **D.caseSignificance**
  - D.caseSignificance = "insensitive"
  - D.caseSignificance = "sensitive"

- **D.typeId**
  - D.typeId = 900000000000003001 |fully specified name|
  - D.typeId = 900000000000013009 |synonym|
  - D.typeId = 900000000000550004 |definition|

- **D.type**
  - D.type = "FSN"
  - D.type = "fullySpecifiedName"
  - D.type = "synonym"
  - D.type = "textDefinition"

- **D.acceptabilityId**
  - D.acceptabilityId = 900000000000549004 |acceptable|
  - D.acceptabilityId = 900000000000548007 |preferred|

- **D.acceptability**
  - D.acceptability = "acceptable"
  - D.acceptability = "preferred"

Additional Syntactic Sugar

- **FSN**
  - FSN = "*heart"
    - D.term = "*heart", D.type = "FSN"
    - D.term = "*heart", D.type = "fullySpecifiedName"
    - D.term = "*heart", D.type = "synonym"
  - FSN = "*heart" LANGUAGE X
    - D.term = "*heart", D.type = "FSN", D.acceptability = * LANGUAGE X
    - D.term = "*heart", D.type = "fullySpecifiedName", D.acceptability = * LANGUAGE X
    - D.term = "*heart", D.type = "synonym", D.acceptability = * LANGUAGE X

- **synonym**
  - synonym = "*heart"
    - D.term = "*heart", D.type = "synonym"
    - D.type = "*heart", D.type = "fullySpecifiedName"
  - synonym = "*heart" LANGUAGE X
    - D.term = "*heart", D.type = "synonym", D.acceptability = * LANGUAGE X
    - D.type = "*heart", D.type = "fullySpecifiedName", D.acceptability = * LANGUAGE X

- **synonymOrFSN**
  - synonymOrFSN = "*heart"
    - synonym = "*heart OR FSN = "*heart"
    - D.type = "*heart", D.type = "fullySpecifiedName"
  - synonymOrFSN = "*heart" LANGUAGE X
    - synonym = "*heart OR FSN = "*heart"
    - D.type = "*heart", D.type = "fullySpecifiedName"

- **textDefinition**
  - textDefinition = "*heart"
    - D.type = "*heart", D.type = "definition"
    - D.term = "*heart", D.type = "fullySpecifiedName"
  - textDefinition = "*heart" LANGUAGE X
    - D.type = "*heart", D.type = "definition"
    - D.type = "*heart", D.type = "fullySpecifiedName"

- **Unacceptable Terms**
  - (D.term = "*heart") MINUS (D.term = "*heart", D.acceptability = " language")
### Language preferences using multiple language reference sets

- LRSs that use the same Language tend to use 'Addition' - i.e. child LRS only includes additional acceptable terms, but can override the preferred term
  - E.g. Regional LRS that adds local dialect to a National LRS
  - E.g. Specialty-specific LRS
  - E.g. Irish LRS that adds local preferences to the en-GB LRS
- LRSs that define a translation to a different language tend to use 'Replacement' - i.e. child LRS replaces set of acceptable and preferred terms for any associated concept
  - E.g. Danish LRS that does a partial translation of the International Release
  - 999999 [Danish language reference set] ELSE [GB English reference set]

<table>
<thead>
<tr>
<th>Other topics</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Any other topics?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Confirm next meeting date /time</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td>The next SLPG meeting will be held in 2 weeks at 20:00 UTC on <strong>Wednesday 6th February</strong>.</td>
<td></td>
</tr>
</tbody>
</table>