

1 Automated Quality Assurance

This document accompanies LIRICS D1.3.

To demonstrate results of the analysis of automated quality assurance, a standard being developed within the LIRICS project has been analysed. The document 'Lexical Markup Framework (LMF)' (at Draft International Standard stage¹) was chosen to show the output obtained from the various stages of the analysis.

1.1 Terminology Lookup

All known terms were annotated including those containing another term. For example, the known term 'object language' contains another known term 'object'. The annotation allows access to the definition for the term. All terms in the specifications 'ISO 1087-1', 'ISO 1087-2', 'ISO 12620' and 'LMF' were annotated. An example of how the terms were annotated in 'LMF' is shown in figure 1.

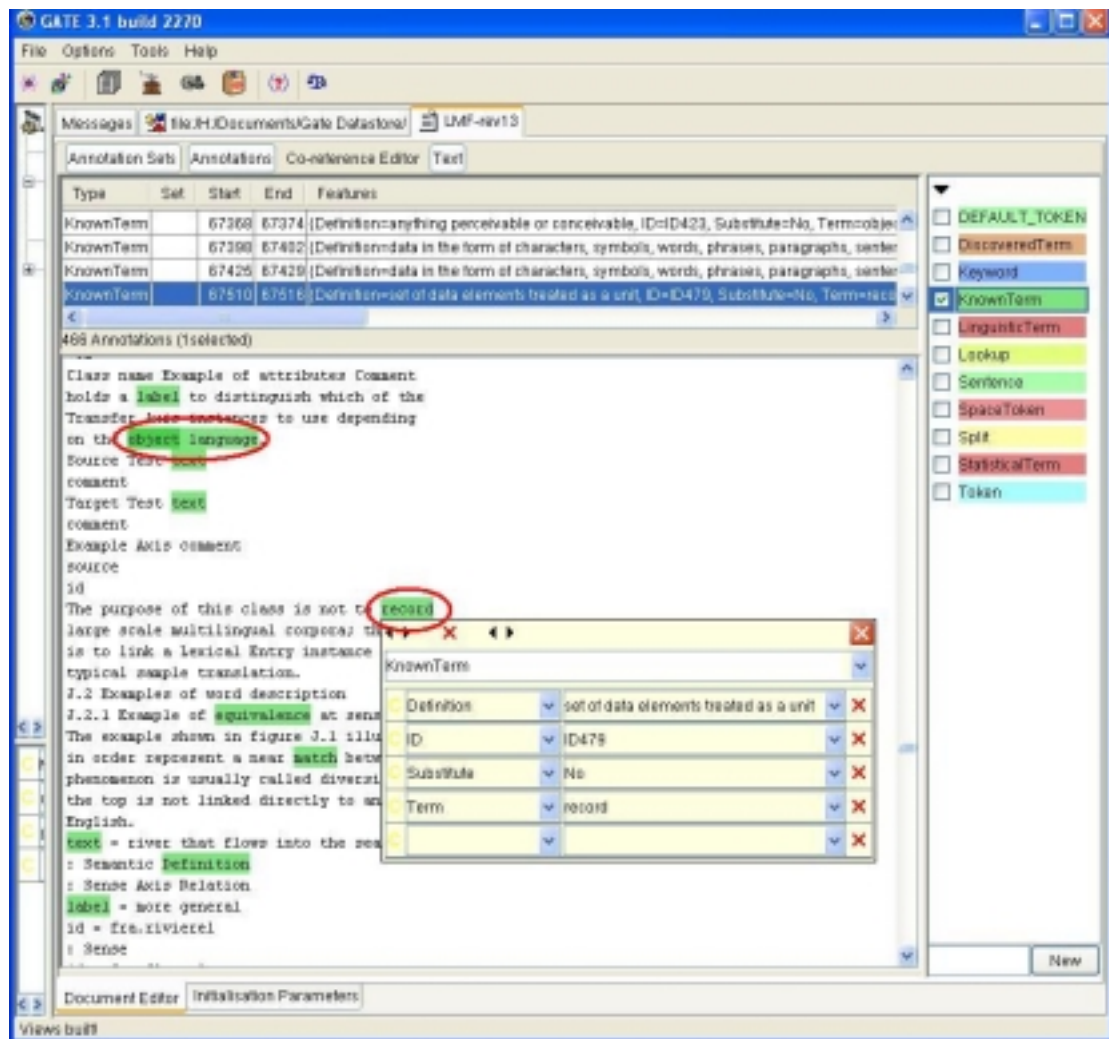


Figure 1: A screenshot of the 'LMF' document in GATE displaying the terminology annotations and a popup window of annotation information for the term 'Record'.

¹ Revision 13, available at: http://lirics.loria.fr/doc_pub/N330_LMF_rev13_For_CD_Ballot.pdf

1.2 Term Finder (Keywords, Statistical and Linguistic)

Similar to Terminology Lookup, a term annotated as a 'DiscoveredTerm' can have annotations within it. For example, in the Figure 4 the potential term 'syntactic annotation' also contains a potential term, 'annotation'. Discovered terms can also have known terms annotated within them. The new term or terms can then be added into the terminology and become part of the annotation set. For example, in the figure below the discovered term 'extension mechanisms' has the existing term 'extension' annotated within it. Similarly the proposed new term 'Unicode string' incorporates the known term 'string'. Decisions over the use of such relationships need to be considered, particularly in the discovered term 'LMF data category selection procedures'. This annotation has the known term 'data category selection' annotated within it, along with the other known terms 'data' and 'data category' within that. An example of how the terms were annotated within GATE for 'LMF' is shown in figure 5.

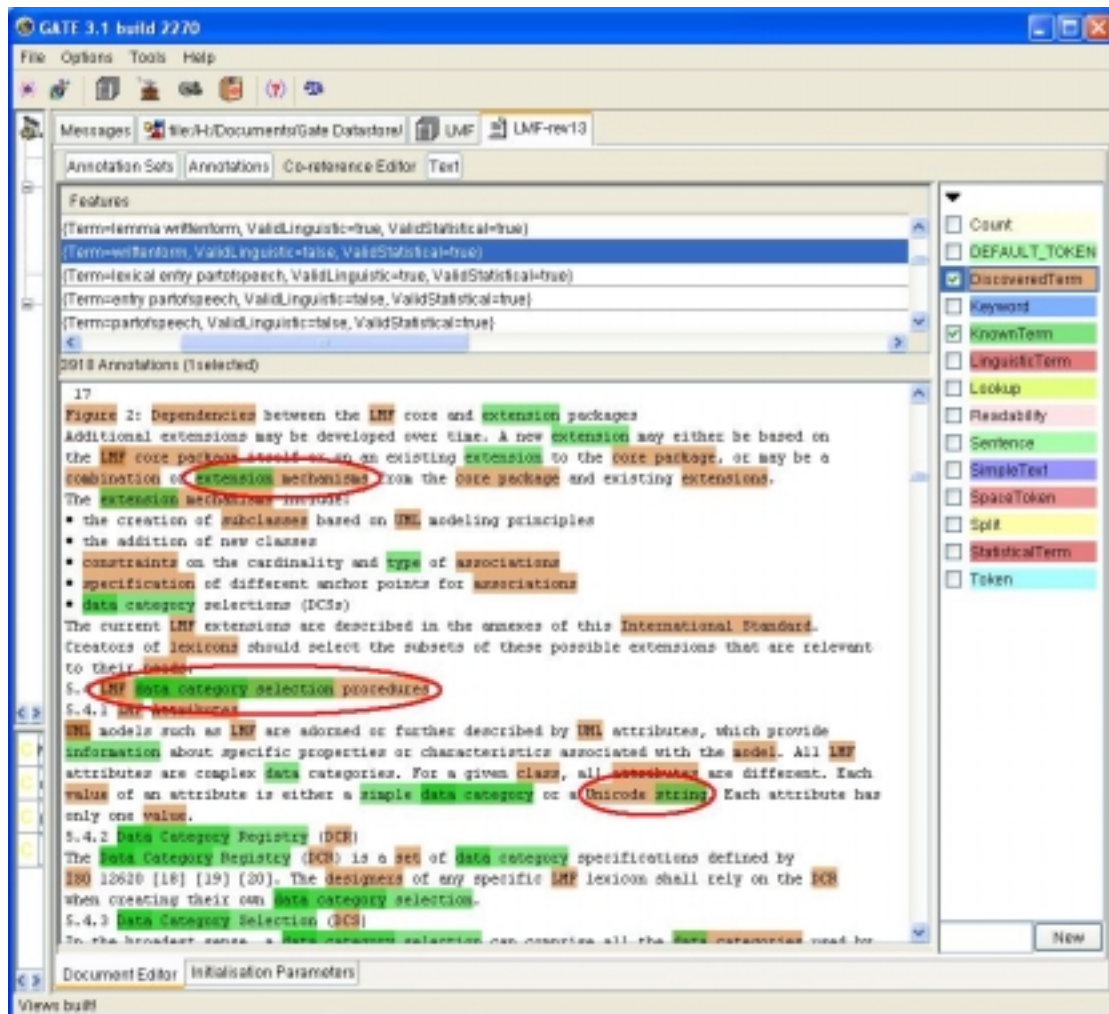


Figure 5: A screenshot of the 'LMF' document in GATE displaying the KnownTerm and DiscoveredTerm annotations.

There were 1380 known terms found in 'LMF' along with 1469 discovered terms. The top 20 known terms in 'LMF' are shown with their frequencies in Table 1.

Term	Count
Class	238
Form	99
Lexical Entry	59
Word	57
Lexicon	46
Data	44
Paradigm	39
Paradigm Class	31
Lemma	30
Extension	27
MWE	27
Note	25
Information	24
Subcategorization Frame	24
Model	22
MRD	19
List	18
Lexeme	17
NLP	17
Data Category	16

Table 1: The top 20 known terms, and their frequencies, in 'LMF'.

Discovered terms were further investigated to evaluate which could be considered as new terms. Terms highlighted by both statistical and linguistic methods could be prioritised for consideration. Further filtering of this list is required, but frequency information can be helpful here also. Examples of discovered terms from LMF are shown in Table 2.

Term	Linguistically Valid	Statistically Valid	Count
example	true	false	58
lmf	false	true	41
iso	true	true	27
subcategorization	false	true	24
language	true	false	23
multilingual	false	true	18
description	true	false	16
sense	true	false	14
sense class	true	false	14
relationship	true	false	13
data categories	true	false	12
verb	true	false	11
subclass	false	true	10
english	true	false	9
inflectional	false	true	9
lexical entry instance	true	false	9
lexicon instance	true	false	8
uml	false	true	8
LMF core package	true	false	8
morphemes	true	false	8
noun	true	false	8
annex	true	false	7

Table 2: Examples of highly frequent discovered terms in 'LMF'.

The discovered terms demonstrated, at lower frequencies, linguistically valid multiword expressions that were surprisingly complex. Examples of such terms are shown in Table 3.

Term	Linguistically Valid	Statistically Valid	Count
Imf data category selection procedures	true	false	2
semantic predicate class section	true	false	2
dual use mrd metamodel	true	false	2
dual use mrd package	true	false	2
subcategorization frame instance	true	false	7
terminological markup framework	true	false	3
semantic definition instance	true	false	2
global information class	true	false	2
LMF conformant lexicon	true	false	2
inflectional paradigm class	true	false	2

Table 3: Examples of potential multiword terms that were discovered in ‘LMF’

Additionally, the linguistic and statistical methods for discovering terms found numerous valid two word expressions that were regularly used. Examples of these are shown in Table 4.

Term	Linguistically Valid	Statistically Valid	Count
sense class	true	false	14
lexicon instance	true	false	8
core package	true	false	6
sense instance	true	false	6
external system	true	false	5
lemma class	true	false	4
narrative description	true	false	4
word forms	true	false	4
affix class	true	false	3
affix slot	true	false	3

Table 4: Examples of frequent discovered two-word terms in ‘LMF’.

There were also notable keywords (single words) identified as valid either linguistically or statistically and frequently used throughout the document. Examples of these are shown in Table 5. Some of these may easily be filtered out.

Term	Linguistically Valid	Statistically Valid	Count
LMF	false	true	34
ISO	true	false	27
subcategorization	false	true	24
multilingual	false	true	18
verb	true	false	11
inflectional	false	true	9
agglutination	true	false	8
UML	false	true	8
noun	true	false	8
sentence	true	false	7

Table 5: Examples of discovered single-word terms in ‘LMF’.

1.3 SimpleText Analyser

The first 150 suggested replacements from an earlier version of the LMF document² were manually analysed with 60 replacements being deemed appropriate. The result of this analysis was sent to the LIRICS project manager and LMF editor to review. 7 of these replacements were deemed appropriate, and the remaining 53 required further considerations to be made. A large proportion of the suggested replacements involved existing terminology. To reduce the number of false positives, the SimpleText analyser was amended, prior to the construction of the Annotation Controller, so that substitutions would no longer be suggested for text already annotated as a known term.

With the revised SimpleText analyser, 'LMF' was analysed again to determine the potential for improvements. A report was produced of new replacements suggested by the SimpleText plug-in for words and phrases deemed unnecessarily complex. The first 350 replacements were analysed manually, with 17 replacements deemed suitable and 14 of these being unique. Every further instance of the substitutions was analysed throughout the rest of the document, 50 instances in total, to see if the replacements were appropriate in every instance. These replacements and their results are detailed in Table 6.

Phrase	Replacement	Appearances In Document	Correct Replacements	%Correct
essential	important	1	1	100.00%
facilitate	help	2	2	100.00%
generally	usually	2	2	100.00%
impossible	not possible	1	1	100.00%
indicated	shown	1	1	100.00%
investigate	examine	1	1	100.00%
latest	last	3	1	33.33%
maintain	keep	2	2	100.00%
needed	necessary	3	3	100.00%
omit	ignore	1	1	100.00%
select	choose	1	1	100.00%
specified	given	14	2	14.29%
various	different	7	5	71.43%
within	in	11	5	45.45%

Table 6: The 14 unique replacements filtered from the initial 350 suggestions with the number of times the replacements were correct throughout the rest of the document.

The 14 occurrences of the word "specified" are detailed in Table 7, which shows the full sentence where the word appeared and whether the replacement "given" was deemed suitable.

Sentence	Valid	Reason
Language identifiers used in LMF-compliant resources shall conform to criteria specified in the ISO 639 family of standards.	Yes	"given" is simpler
Script identifiers used in LMF-compliant resources shall conform to criteria specified in the ISO 15924 Codes for Script Identification.	Yes	"given" is simpler
In a multilingual configuration, the external linking is represented by a Multilingual External Ref instance, as specified in the NLP multilingual notation package.	No	Loss of accuracy
In a monolingual configuration, the external linking is represented by a Monolingual External Ref instance, as	No	Loss of accuracy

² Revision 9, available at: http://lirics.loria.fr/doc_pub/LMF%20rev9%2015March2006.pdf

specified in NLP semantic package.		
Note: the mechanisms for an intensional description of the morphology are specified in the Paradigm class annex.	No	Loss of accuracy
A noun phrase: somebody, that is not fully specified in the sense that the only restriction that is expressed is that the nucleus of the phrase must be of type /human/.	No	Not the same meaning
The objective of the whole package being to provide a generic representation of MWE combinations within a give language, the components are not referenced directly but on the contrary, they are referenced by their respective ordering as specified in the List Of Component instance.	No	Loss of accuracy
Example: An attribute Valuation instance may be the pair partOfSpeech and adjective. Note: the class name of this attribute is not specified .	No	Loss of accuracy
The Sense class is specified in the core package.	No	Loss of accuracy
In this process, lexicon developers shall use the classes that are specified in the LMF core package (section 5.2).	No	Loss of accuracy
The dependencies of the various extensions are specified in the following diagram.	No	Loss of accuracy
A fully specified verb: throw, referenced by rank one within the List Of Components instance.	No	Not the same meaning
A fully specified second noun phrase to the lions referenced by ranks one, two and three within the List Of Components instances.	No	Not the same meaning
Developers shall define a data category selection (DCS) as specified for LMF data category selection procedures.	No	Loss of accuracy

Table 7: The descriptions detailing when “given” was a suitable replacement for “specified”.

The 11 occurrences of the word “within” are detailed in Table 8, which shows the full sentence where the word appeared and whether the replacement “in” was deemed suitable.

Sentence	Valid	Reason
This Paradigm Class instance has a name but is not analytically described within the lexicon.	Yes	“in” is simpler
A fully specified verb: throw, referenced by rank one within the List Of Components instance;	Yes	“in” is simpler
Within the current International Standard, the first part may give a Semantic Definition instance and the two last parts may give two Statement instances.	Yes	“in” is simpler
A fully specified second noun phrase to the lions referenced by ranks one, two and three within the List Of Components instances. This prepositional phrase is labelled as /plural/.	Yes	“in” is simpler
The objective of the whole package being to provide a generic representation of MWE combinations within a give language, the components are not referenced directly but on the contrary, they are referenced by their respective ordering as specified in the List Of Component instance.	Yes	“in” is simpler
form that a word can take when used in a sentence or a phrase within an inflectional language.	No	Loss of accuracy
form that a word can take when used in a sentence or a phrase within an agglutinating language.	No	Loss of accuracy
Synset is a class representing a common and shared meaning within the same language.	No	Loss of accuracy
Example: when applied to an inflectional languages, "extensional" means that all inflected forms will be explicitly described within one Lexicon instance.	No	Loss of accuracy

This approach represents stems, rules and conditions within the lexicon but an external parser is needed to fully interpret the rules.	No	Loss of accuracy
Lexicon is a class containing all the lexical entries of a given language within the entire resource.	No	Loss of accuracy

Table 8: The descriptions detailing when “in” was a suitable replacement for “within”.

The 7 occurrences of the word “various” are detailed in Table 9, which shows where the word appeared and whether the replacement “different” was deemed suitable.

Sentence	Valid	Reason
The dependencies of the various extensions are specified in the following diagram.	Yes	“different” is simpler
Lexicon designers can freely structure the various axes directly or indirectly between and among different languages.	Yes	“different” is simpler
The SynArgMap is a class representing the relationship that maps various Syntactic Argument instances of the same Subcategorization Frame Set instance.	Yes	“different” is simpler
The larger the number languages and the number of links, the greater the chance that lateral links between the various languages can prove faulty.	Yes	“different” is simpler
A Subcategorization Frame Set groups various syntactic constructions that appear frequently for certain sets of words.	Yes	“different” is simpler
In these latest examples, each form is defined from the lemma (or one stem) of the entry with various operations like adding affixes.	No	Loss of accuracy
set of form operations that build the various forms of a lexeme, possibly by inflection, agglutination, compounding or derivation.	No	Not the same meaning

Table 9: The descriptions detailing when “different” was a suitable replacement for “various”.

The 3 occurrences of the word “latest” are detailed in Table 10, which shows the full sentence where the word appeared and whether the replacement “last” was deemed suitable.

Sentence	Valid	Reason
For undated references, the latest edition of the normative document referred to applies.	Yes	“last” is simpler
Component is a class representing a reference to a lexical entry when this latest one is an element of List Of Component class.	No	Loss of accuracy
In these latest examples, each form is defined from the lemma (or one stem) of the entry with various operations like adding affixes.	No	Doesn’t quite make sense

Table 10: The descriptions of when “last” was a suitable replacement for “latest”.

1.4 Readability analysis

Some SimpleText replacements were appropriate in every further instance such as “needed”, “facilitate”, “generally” and “maintain”. However other words rarely had correct replacements, in particular ‘latest’ was never suitable again. The majority of proposed SimpleText replacements were found not to be suitable and leave much room for investigation of how to focus the substitutions more accurately. To investigate the extent that this limited number of substitutions would influence the readability scores of the document the Replacer plug-in was run. All the readability scores were slightly reduced except for the FOG and SMOG results which increased a little. The increase in readability scores can be attributed to the fact that some SimpleText replacements do not decrease readability scores. In fact the number of words in a document can actually *increase* due to some of the replacements. An example of this occurrence is the substitution of “impossible” for “not possible”. One culprit of the increase is probably “necessary” replacing “needed”. The replacement has more syllables than the word it is replacing, which would increment the number of complex words in the document.

This would explain why only FOG and SMOG have increased: these readability measures count polysyllable words as complex. Other replacements such as “important” for “essential” have no effect on readability scores. Readability scores before and after the replacements are shown in Table 11. These tests were performed with the known terms annotated in the document which influence the complex word count in the FOG and SMOG readability scores.

Score	Before	After
Kincaid	14.158	14.153
Flesch	33.977	34.059
FOG	15.007	15.032
SMOG	13.608	13.626
ARI	13.824	13.817

Table 11: Readability scores before and after the SimpleText process.

FOG and SMOG were tested in two forms: first in “raw” form (Raw), second by removing known terms from the calculations (Surrey). Any word of more than 3 syllables which was part of a known term was not considered as a complex word in these calculations since it is assumed that inclusion in the terminology has already accounted for readability. This approach improves readability scores by reducing the number of complex words. The FOG and SMOG scores before and after the replacements, and omitting or including complex words, are shown in Table 12.

Score	Before Replacements		After Replacements	
	Raw	Surrey	Raw	Surrey
FOG	16.692	15.007	16.716	15.032
SMOG	15.003	13.608	15.021	13.626

Table 12: Readability scores omitting and utilising terminology before and after the SimpleText process.