D9.1 MOLTO test criteria, methods and schedule

http://www.molto-project.eu

Contract No.:	FP7-ICT-247914
Project full title:	MOLTO - Multilingual Online Translation
Deliverable:	D9.1. MOLTO test criteria, methods and schedule
Security (distribution level):	Confidential
Contractual date of delivery:	M7
Actual date of delivery:	October 2010
Type:	Report
Status & version:	Final
Author(s):	L. Carlson et al.
Task responsible:	UHEL
Other contributors:	

ABSTRACT

The present paper is the summary of deliverable D 9.1 as of M6.

This deliverable is to define the requirements for both the generic tools and the case studies in a coherent way that can lead to maximal synergy between work packages. To do this, we need to detail the project plan and schedule. This then implies the main outline of the evaluation schedule.

This paper is structured into an introduction followed by sections per work package, The WPs are divided into the front end WPs (WP3 and use cases) and the back end ones (WPs 2,4,5). For each WP we survey promises from DoW, ongoing work, and derive requirements from them, followed by evaluation plans or recommendations. Text in brackets refer to source. Action points are in boldface.

The wealth of cited content aims to bring different strains of documented work planned or in progress together, in order to get an updated view of the ongoing MOLTO process, and thus cover the bases for making the tool and user WP requirements meet. We take as base what the technology offers and scale user expectations from that.

MOLTO Deliverable 9.1

Table of Contents

Introduction

The present paper is the summary of deliverable 9.1 as of M6. Work package materials can be found at the UHEL MOLTO website (https://kitwiki.csc.fi/twiki/bin/view/MOLTO). This document also links to the MOLTO official website (http://www.molto-project.eu/).

(The official MOLTO website is the prime place for coordinating the project as (long as) material on it is uncluttered, reliable and up to date. For local work, informal project communication and creative planning, the UHEL MOLTO website is open to all MOLTO partners.)

This paper is structured into an introduction followed by sections per work package, The WPs are divided into the front end WPs (WP3 and use cases) and the back end ones (WPs 2,4,5). For each WP we survey promises from DoW, ongoing work, and derive requirements from them, followed by evaluation plans or recommendations. Text in brackets refer to source. Action points are in boldface.

The wealth of cited content aims to bring different strains of documented work planned or in progress together, in order to get an updated view of the ongoing MOLTO process, and thus cover the bases for making the tool and user WP requirements meet. We take as base what the technology offers and scale user expectations from that.

The deliverables from WP 9 Requirements and Evaluation are:

Del. no	Del. title	Nature	Date
D 9.1	MOLTO test criteria, methods and schedule	R	M6
D 9.2	MOLTO evaluation and assessment report	R,M	M36

The place of WP9 in MOLTO's overall strategy is stated here.

`[DoW 14 MOLTO overall strategy and general description]`

> *The project will develop tools and applications in parallel.* ***The leading idea is to have working prototypes from the beginning, and deliver updates frequently.*** *The work is divided into four kinds of packages:*
> *• Management and dissemination: WP1 and WP10. These run throughout the project.*
> *• Generic tools: WP2–5. These start early in the project.*
> *• Case studies: WP6–8. These start later than the tools, because they assume some maturity of the tools. However, some of them also involve data collection, which can be started earlier.*
> *•* ***Requirements and evaluation: WP9****. This runs throughout the project.* ***Its purpose in the beginning is to define the requirements for both the generic tools and the case studies*** *in a coherent way that can lead to maximal synergy between work packages, (the case studies are otherwise independent of each other).* ***Later in the project, WP9 performs evaluation and delivers feedback.*** *In the last phase of the project, when the development of new functionalities in tools and case studies has stopped (month 30),* ***WP9 takes care of bug fixing and consolidation of the tools and case studies****, so that everything remains coherent.*

We go over the later WP9 tasks first:

- **performing evaluation**: individual partners as contributors to WP9 perform WP-wise evaluation subject to criteria collected in this document. The WP10 leader collects the contributions to be included for the periodic reports as required.
- **bug fixing and consolidation**: The WP leader and/or the coordinator will maintain bug tracking tools and distribute bug reports or feature requests to the appropriate partner(s).

Deliverable 9.1 is to define the requirements for both the generic tools and the case studies in a coherent way that can lead to maximal synergy between work packages. To do this, we need to detail the project plan and schedule. This then implies the main outline of the evaluation schedule.

Schedule

The MOLTO dependency chart only shows dependencies for WP 9 with the use cases WP 6-8 plus the dissemination WP 10. The boldfaced bits above entail that there are dependencies to the tools work packages as well.

By the MOLTO timetable, WPs 2,4,9 (tools, ontology, req/eval) started at once. Translation tools WP3 and use case WPs 6 and 8 start at m7 (Varna). Patents use case WP7 has not started due to failure of partner.

By the DoW, MOLTO aims to have working prototypes on the way. So far, each partner has been providing their own demos. Progressively, there will be more need for integration, WP3 in particular will use most of the rest as components. In the best case, integration can be just plugging in APIs, with local bilateral negotiation at best between a provider and a user. But to ensure this, we must agree in time what the APIs will provide.

As suggested in the DoW text (but not spelled out in the schedule), **specification/version checkpoints should be agreed more often between the tools WPs.** At Varna, we get the first update of the tools and ontology work packages**.** It also helps to be clear just what capabilities each release is planned to offer. Proposals what to insert into the project schedule are made along the way below. Checkpoints can be constructed from the deliverables list and the milestone table (in DoW). The deliverables list implies these checkpoints with implications to the evaluation timetable:

• M03 MOLTO web service, first version – we got the phrasebook running from MOLTO website.

• M12 GF grammar compiler API – what does this add?

• M12 Web-based translation tool available – Krasimir Angelov is making a new web-based GF editor, which will be published as the MS3.

• M18 Grammar IDE, Ontology-GF interface, Translation tools API – a web translation platform that allows on the fly extension of lexicon and grammars with ontology tools. Evaluation of lexicon and grammar extension can start.

• M24 Translation tool prototype running – translator tool evaluation can start with test users (on the museum and if relevant the mathematics case).

• M30 GF tools integrated with SMT tools – evaluating the combined system with the patent use case can start. Manuals done: testing with new users possible

Having cleared the schedule, we go through the WP 9.1 tasks boldfaced from the WP9 statement of purpose.

[DoW WP9]

The work will start with ***collecting user requirements for the grammar development IDE (WP2), translation tools (WP3), and the use cases (WP6-8). We will define the evaluation criteria and schedule in synchrony with the WP plans (D9.1).*** *We will define and collect corpora including diagnostic and evaluation sets, the former, to improve translation quality on the way, and the latter to evaluate final results.*

Collecting user requirements

We have not been able to do much interviewing here because the patent user partner (WP7) is missing and the two others have not started their WPs yet. We have not got real end users in the use cases. In the mathematics case, the end users could be math teaching platform developers; in the patent case, patent office staff; in the museum case, museum workers. These are content professionals with more than average technical facility.

The use cases were scheduled as follows.

WP6 Case Study: Mathematics	Start month 7
WP7 Case Study: Patents	Start month 4 (not started due to loss of patent partner)
WP8 Case Study: Cultural heritage	Start month 13

This problem was implicit in the original timetable which expected WP9 to work on the use cases before the use case WPs started working. This was noted in the kickoff meeting and agreed that this task would be rescheduled as necessary.

Pending user input, we decided to derive requirements from MOLTO's promises and compare them to the tools resources. The promises made by MOLTO from DoW are summarised below.

[DoW 5 Progress beyond the state of the art]

The single most important S&T innovation of MOLTO will be ***a mature system for multilingual on-line translation, scalable to new languages and new application domains.***

The single most important ***tangible product of MOLTO is a software toolkit, available via the MOLTO website****. The toolkit is a family of open-source software products:*

1. a grammar development tool, available as an IDE and an API, to enable the ***use as a plug-in to web browsers, translation tools, etc, for easy construction and improvement of translation systems and the integration of ontologies with grammars*** (WP2)
2. ***a translator's tool, available as an API and some interfaces in web browsers and translation tools*** (WP3)
3. a grammar library for linguistic resources
4. a grammar library for the domains of mathematics, patents, and cultural heritage

Defining evaluation criteria

A helpful list of quality dimensions relevant to MOLTO evaluation can be derived from the DoW list of links between the main objectives and the tasks in WPs:

1. adaptability of translation systems: WP2
2. user friendliness and integration in workflows: WP3
3. integration with semantic web technology: WP4
4. usefulness on different domains: WP6, WP7, WP8
5. scaling up towards more open text: WP5, WP7
6. quality of translation: WP9
7. wide user adaptation and exploitability: WP10

Here are some measurable expected outcomes from DoW. Most of them are directly applicable as testable quantitative evaluation measures. It is another thing how many test rounds we can do, given the need of fresh test subjects.

Feature	***Current***	***Projected***
Languages	*up to 7*	*up to 15*
Domain size	*100's of words*	*1000's of words*
Robustness	*none*	*open-text capability*
Development per domain	*months*	*days*
Development per language	*days*	*hours*
Learning (grammarians)	*weeks*	*days*
Learning (authors)	*days*	*hours*

1. languages treated simultaneously: up to 15
2. domains with substantial applications: 4 (NOTE: "substantial" not quantified here)
3. translation quality: "complete" or "useful" on the TAUS scale (Translation Automation Users Society)
4. source authoring: the MOLTO tool for writing translatable controlled text can be learned in less than one hour, ***the speed of writing translatable controlled text is in the same order of magnitude as writing unlimited plain text***

The number 18 of grammar library languages is the minimum number of languages we expect to be available at the end of MOLTO. ***The number 3 to 15 is the number of languages actually implemented in MOLTO's domain grammars (3 in WP7, 15 in WP6 and WP8)****.*

The measurements of all these features are performed within WP9 in connection to the project milestones.

The advisory group will confirm the adequacy and accuracy of the measurements.

The objects of evaluation – even the translated texts – vary considerably per WP. We detail some criteria per WP below. Evaluation criteria and methods have been collected on the UHEL MOLTO website (esp. Evaluation Cookbook).

Defining evaluation corpora and tools

Not much could be done here (yet). We have not got patent corpora. The mathematicians have yet to collect their word problems. We got a small museum text corpus (approx. 25000 words in Swedish, a set of 9 short passages translated into English presumably by non-native speakers) from Gothenburg.

We have translated parts of this corpus both manually and using MT for test material in BLEU evaluation. A pilot comparing BLEU scores on this material to a manual error analysis is on the way.

We have found time to install an evaluation platform, collect and test standard issue translation quality evaluation tools, to develop forthcoming MOLTO lexicon tools, to learn GF and develop ideas about the ontology to grammar interface. The IQmt evaluation platform was tested on a small sample of machine and human translated text from English to Finnish (see UHEL wiki: Evaluation Cookbook). The results suggest that statistical evaluation methods are not suitable for MOLTO.

UHEL also took part in the MOLTO phrasebook task, a demo for translating touristic phrases between 14 European languages: Bulgarian, Catalan, Danish, Dutch, English, Finnish, French, German, Italian, Norwegian, Polish, Romanian, Spanish, Swedish. This experiment presents one way evaluate the effort required for adding new language versions (more on this in section about WP2).

We divide the rest of the paper by WPs into the front end: translation tool, the use cases and associated lingware (ontologies and grammars), and the back end: the translation system (WPs 2,4,5), presented in this order. We also try to form an idea about what WPs are currently about to see how they are construing their tasks. Information about this (at least task titles) was found on MOLTO website.

Front end

WP3 translator's tools

MOLTO translation scenario and user roles

The MOLTO workflow is a break to tradition in the professional translation business as well as the consumer end in that it merges the roles of content author and translator. In professional translation, a document is authored at source and the translator's work on the source is read-only. At the consumer end, MT is largely used for gisting from unknown languages to familiar ones.

The MOLTO change of roles will also entail a change of scenarios. Since the MOLTO scenario implies major differences to the received translation workflow and current roles and requirements from translation client, translator, revisor etc. MOLTO is not likely to impact translation business at large in the near future. Instead, it has its chances in entering and creating new workflows, in particular, in **multilingual web publishing**. Multilingual websites are currently developed by means of crowdsourcing translation (Schonfeld 2009) or professional translation with tools borrowed from the software localization business ("Drupal Translation Management"). MOLTO could complement or replace this workflow with its new role cast of a content producer or technical editor that generates multilingual content from a single language source. Applications may include multilingual Wikipedia articles, e-commerce sites, medical treatment recommendations, tourist phrasebooks, social media, SMS.

As for CAT in general, the advantages of MOLTO can be particularly clear in versioning of already existing sites. Once MOLTO has been primed for one text it can translate any number of (sufficiently) similar ones, such as updates for already translated texts.

We next review user requirements by type of user and the expected expertise of each. Consider the role cast around MOLTO. The role cast in MOLTO can have at least these:

•Author
•Editor
•Translator
•Checker
•Ontologist
•Terminologist
•Grammarian
•Engineer

So far, all of these roles are merged. Different use scenarios may separate some and merge others. Peculiar to MOLTO is the merge of the author/editor/translator roles. In the MOLTO scenario, the editor-translators cannot be expected to know (all) the target language(s). The target checker(s) and terminologist(s)-grammarian(s) are likely to be different from them, possibly a widely distributed crowd.

The translator's tool serves primarily for author/editor/translator/checker roles. It links to TF which serves ontologist/terminologist roles (and connects them to the former). Presumably, the Grammar IDE supports the last four roles on the above list.

The author is likely to be some sort of an expert on the subject matter, but not necessarily an expert on ontology work. The editor, if separate from the author, could be less of a subject expert but possibly more of an ontologist. How much of a difference there need be between these roles depends on the cleverness of the MOLTO tools.

Say an author types away and MOLTO counters with questions caused by the underlying ontology (of type "do you mean this or that?") Unless the author agrees with the ontology, s/he may be hard put to answer, while an editor/ontologist (familiar with the ontology and/or the way MOLTO works) may know how to proceed – to choose the right thing or to realize the right alternative is missing and how to fix it.

Analogous comments can be made of the relations between author, translator, checker and terminologist. It is all very well for the author to immediately see translations in umpteen languages s/he does not know. S/he has no way of knowing whether they are correct (unless MOLTO provides some way to check – say back translation with paraphrase?). Also, concrete grammars may ask awkward questions (of the type do you mean male or female, familiar or polite?). To get things right, the author would need to know whether one should be familiar or polite in language N. Here, s/he needs (to be) a translator or native checker. Considerations like this need to be taken into account in WP3 requirements analysis.

WP3 requirements

The following points, taken from section 9 in DoW, recap the main ingredients of the translation tools made available to WP3 by WP2.

- Predictive parsing – yields word predictions to guide the author
- Syntax editing – when a noun is changed, agreement changes accordingly
- Solving ambiguity – ranking by statistical models or manual disambiguation
- No need to install anything – MOLTO tools will work as plug-ins to ordinary tools such as web browsers and text editors
- Dealing with unrestricted output – the user can extend the grammar or use SMT as fallback
- Lexicon extension and example-based grammar writing (more on this later)

The three use cases probably will not use one and the same platform. It is not even sure they want the same features. The mathematicians are likely to need some math editing tool and perhaps access to a computational algebra solver. Patent translators may need access to patent corpora and databases. Museum people may need to work with images. Future MOLTO users may have their own favourite platforms with such facilities in place.

Rather, the WP3 translation tools deliverable should be a set of plugins usable in many different platforms. Still, we need a flagship demonstrator for the project. The flagship demonstrator should be a generic web editing platform. Minimally, it can be an extension of the existing GF web translation demo. In the best case, it could be installed as a set of plugins to some existing web platform like Mediawiki, Drupal and/or some open source CAT tool(s).

The demonstrator should be able to have at least the following plugins:

- TM (translation memory)

• Statistical translator (if separate from above)
• GF translation editor (including autocompletion and syntax editing)
• GF grammar IDE
• TF ontology/lexicon manager
• Ontotext ontology tools (if separate from above)

All or parts of some existing web translation platform(s) could be taken as starting point. Or conversely, some existing CAT tool components could be plugged into ours.

Deliverables

Del. no	Del. title	Nature	Date
D 3.1	MOLTO translation tools API	P	M18
D 3.2	MOLTO translation tools prototype	P	M24
D 3.3	MOLTO translation tools / workflow manual	RP, Main	M30

The MOLTO worfklow and role play must be spelled out in the grammar tool manual (D 2.3) and the MOLTO translation tools / workflow manual (D 3.3). **We should start writing these manuals now, to fix and share our ideas about the user interfaces.**

WP3 evaluation

The main claims to fame in MOLTO are to produce high automatic translation quality, particularly in view of faithfulness, into multiple languages from one pre-editable source, and as a way to that, practically (= economically) feasible multilingual online translation editing with a minimum of training; as promised in DoW, the author should be able to learn to use MOLTO tools within hours. These claims should then be among the items to evaluate.

Quantified evaluation of translation tool features starts when the translation tool prototype developed in WP3 is ready (M24). The tests can be developed and calibrated on the initial demonstrator at M18. We distinguish below between evaluating the translation result and evaluating the translation process.

3a. Evaluating the translation result

We argue below that there is little sense for WP9 to quantitatively measure MOLTO translation quality with standard MT evaluation tools except at the end of MOLTO (D9.2). On the way there, WPs (in particular the GF grammar and SMT WPs) should institute their own progress evaluation schedules. They may then outsource translation quality evaluations to WP9 when appropriate. What we want to avoid is an externally imposed evaluation drill during WP work which can produce skewed results and cause useless delays on the way.

We have created a UHEL MOLTO TWiki website to coordinate our work packages internally. The website is open for other MOLTO partners as well. We have installed standard SMT evaluation tools on our local test platform (hippu.csc.fi). A pilot study on measuring translation fidelity has been conducted

in a PhD project associated with MOLTO (Maarit Koponen).

We promised in the DoW about translation quality assessment ("WP9: User Requirements and Evaluation") that we would use both automatic metrics (e.g. BLEU) and TAUS quality criteria. More weight is given to non-automatic metrics, because the statistical methods are not well suited for measuring semantic fidelity. The criteria mentioned in DoW (scalability, portability, and usability) mean that MOLTO should have wider coverage, be easier to extend and need less expertise than similar (symbolic, grammar-based, interlingual) solutions heretofore.

Applying BLEU and similar methods which compare MT output to human model translations promises to be laborious in the case of MOLTO because we have a large number of less-common target languages and lack use case related corpora. Though we do not have full knowledge yet what corpora we shall have access to, they are not likely to provide a wealth of (preferably many parallel) human model translations for comparison in the special domains we have:

• We expect the mathematics WP to involve a small number (tens or hundreds) of one-paragraph examples

• The museum corpus (at least so far) is not much larger (25K words in all). The largest subset is Swedish only.

• We do not know yet what to expect from the patent partner.

The main difficulties for automatic comparison measures are ambiguities in natural languages: Usually, there is more than one correct translation for a source sentence; there are ambiguities in the choice of synonyms as well as in the order of the words. Allowance for free variation through synonymy and paraphrase (free translation in general) is made with more comparison text. For instance, the NIST evaluation campaign uses four parallel translations (to the same language) of texts in the order of 15-20K words.

What is more to the point, BLEU results are not likely to prove MOLTO's strengths, because they are not sensitive to fidelity, being in this respect like the n-gram SMT methods they simplify. Preliminary tests to this effect have been conducted by Koponen (2010).

BLEU and similar tests have been developed in the context of SMT and for the assimilation (gisting) scenario. Most of the weight in BLEU or WER like measures comes from matched words and shorter n-grams. These measures point in the right direction as long as translation quality is low (as long as long distance dependencies and fidelity do not matter).

Human evaluation measures distinguish between fluency and fidelity, but no such distinction is not made for automatic evaluation measures, which commonly judge the overall quality of a candidate sentence or system. Leusch (2005) shows that some measures have preferences for certain aspects – the unigram PER correlates with adequacy to a higher degree than the bigram PER, whereas this is vice versa on the fluency, but the observation remains to be exploited.

To evaluate fidelity as well as fluency, more grammar sensitive measures are needed. In smaller use cases, human evaluation is likely to be the cost effective solution ("The cost of human reference translations"). An innovative approach suggested by work in Koponen (to appear) would to develop the MOLTO evaluation methodology using MOLTO's own technology. The idea is to use simplified (MOLTO or other) parsing grammars to test fidelity and domain ontologies to test fluency.

Fidelity (preservation of grammatical relations) would be gauged by using simplified grammars to parse summaries of text and comparing MOLTO translations of summaries with summaries of

translations. The assumption is (like it implicitly is in BLEU) that the translator is more reliable with shorter bits (and there are more of them).

Acceptability of lexical variation in the target text would be checked (not against parallel human translations but) against multilingual domain ontologies (e.g., use vessel or boat instead of ship).

Note the analogy here to BLEU's use of n-grams as a simplification of SMT methods to compare SMT to human targets. Work developing these ideas is in progress in a PhD project associated to MOLTO (Koponen to appear). The planned GF/SMT hybrid system is interesting here. It suggests analogous ideas for hybridizing statistical and grammar based evaluation measures.

At the evaluation phase towards the end of MOLTO, a comparison of (say) the patent case output to competing methods using generic tools like the SMT evaluation tools and TAUS criteria is worth doing, and has been promised in the DoW. On the way there, however, we prefer developing and applying MOLTO specific evaluation methods.

3b. Evaluating the translation process

WP9 aims to set requirements and evaluate the MOLTO translation workflow from the beginning. We argue below that evaluating the translation workflow and translator productivity are particularly important in MOLTO. For related work in other projects, see (UHEL wiki: Evaluation Cookbook). Our initial proposals follow below.

The MOLTO pre-editing strategy lets an author or technical editor modify the text, the translator enrich the vocabulary, and the grammarians perfect the grammar until the translation result is acceptable. Therefore the success criterion for the MOLTO approach must be how much effort it takes to get a translation from initial state to a break-even point (as defined by the use case). A translation can always be made better with more work on the tool, but the crux is when the result pays the effort. The DoW sets these quantitative expectations on source editing:

```
[DoW  5 Progress beyond the state of the art]
```

> *Source authoring: the MOLTO tool for writing translatable controlled text can be learned in less than one hour,* ***the speed of writing translatable controlled text is in the same order of magnitude as writing unlimited plain text***

"Of the same order" mathematically means that writing with MOLTO is not ten times slower than writing without it. We should clock this.

We pick up this discussion again under WP2 in connection with measuring the vocabulary and grammar extension effort.

WP6 Case Study: Mathematics

Objectives

The description of this case study in DoW and in the MOLTO website ("WP6 Case Study: Mathematics") makes apparent that the math use case demonstrator is not so much a translation editor as natural language front end to computer algebra.

The impression is confirmed by an email from Jordi Saludes:

> *"The simplest implementation will be a terminal-based question/answer system like ELIZA, but focused on solving word problems. It will start by giving the statement of the problem, then it will do computations for the student/user, list unknowns, list relations between unknowns, state the progress of the resolution and, maybe, give hints.*
>
> *We are thinking about the kind of word problems which require solving a system of (typically two) linear equations. In Spain these are addressed to first or second year high school students."*

Deliverables

Del. no	**Del. title**	**Nature**	**Date**
D 6.1	Simple drill grammar library	P	M15
D 6.2	Prototype of commanding CAS	P	M23
D 6.3	Assistant for solving word problems	P,Main	M30

On the way to the demonstrator, the plan is to devise small ontologies describing math word problems and verbalise them using the MOLTO platform and WebAlt project math GF grammars. This phase of the work can be evaluated on the lines indicated under WP3 above.

WP7 Case Study: Patents

Objectives

As stated in the DoW, the objectives of this WP are three:

(i)to create a prototype of a system for MT and retrieval of patents in the bio-medical and pharmaceutical domains,

(ii)allowing translation of patent abstracts and claims in at least 3 languages, and

(iii) exposing several cross-language retrieval paradigms on top of them.

Evaluation

The final prototype will be tested and evaluated according to the general criteria in terms of usability and translation quality. For translation quality, the methodology will be that of WP5, that is a combination of automatic metrics (lexical, semantic and syntactic metrics as defined within the IQmt package) and human evaluation.

Usability should be evaluated within a real system. The prototype will be shared with EPO in order to examine its feasibility as a part of a patent retrieval system

Finally, there is another EU project about translating patents. One way to assess MOLTO could be to compare our results to them:

```
[ PLuTO  Innovation description
  http://cordis.europa.eu/fp7/ict/language-technologies/project-pluto_en.html]
```

> *PLuTO will develop a rapid solution for patent search and translation by integrating a number of existing components and adapting them to the relevant domains and languages. CNGL bring to the target platform a state-of-the-art translation engine, MaTrEx, which exploits hybrid statistical, example-based and hierarchical techniques and has demonstrated high quality translation performance in a number of recent evaluation campaigns. ESTeam contributes a comprehensive translation software environment to the project, including server-based, multi-layered, multi-domain translation memory technology. Information retrieval expertise is provided by the IRF which also provides access to its data on patent search use-cases and a large scale, multilingual patent repository. PLuTO will also exploit the use-case holistic machine translation expertise of Cross Language, who have significant experience in the evaluation of machine translation, while WON will be directly involved in all phases of development, providing valuable user feedback. The consortium also intends to collaborate closely with the European Patent Office in order to profit from their experience in this area.*

The latter option implies the agreement and interaction between PLuTO and MOLTO.

Deliverables

Del. no	**Del. title**	**Nature**	**Date**
D 7.1	Patent MT and Retrieval Prototype Beta	P	M21
D 7.2	Patent MT and Retrieval Prototype	P	M27
D 7.3	Patent Case Study Final Report	RP, Main	M33

WP8 Case Study: Cultural heritage

Objectives

According to DoW, the objective of WP8 is to build an ontology-based grammar for museum information, which will enable descriptions of museum objects and answering to queries written in natural language. The ontology used by Gothenburg City Museum is CIDOC Conceptual Reference Model (CRM), a high-level ontology to enable information integration for cultural heritage data and their correlation with library and archive information. Ontotext has created a GF grammar which translates natural language queries into SPARQL queries, which will be used to provide a museum visitor means to write queries about museum objects in his/her own language.

The CIDOC CRM analyses the common conceptualizations behind data and metadata structures to support data transformation, mediation and merging. It is property-centric, in contrast to terminological systems. It is now in a very stable form, and contains 80 classes and 130 properties, both arranged in multiple isA hierarchies.

Semantic Computing Research Group (SeCo, Eero Hyvönen) has an Ontology for museum domain (MAO). MAO is an ontology for the museum domain, used for describing content such as museum items. MAO is ontologically mapped to the Finnish General Upper Ontology YSO and has been created as part of the FinnONTO-project. The most important application of MAO is The Semantic Portal for Finnish Culture Kulttuurisampo. Seco is specialised in indexing websites with ontologies. They are currently translating their ontologies into Finnish and Swedish.

Deliverables

Del. no	Del. title	Nature	Date
D 8.1	Ontology and corpus study of the cultural heritage domain	O	M18
D 8.2	Multilingual grammar for museum object descriptions	P	M24
D 8.3	Translation and retrieval system for museum object descriptions	P,Main	M30

We will evaluate deliverables D8.2 and D8.3 with respect to both usability and translation/NLG quality. As argued in section WP3 Evaluation, human evaluation will be weighed over automatic evaluation measures.

The back end

The back end, relative to the translation tool and user cases, consists of the GF grammar developer's tools, the ontology support, and the SMT facility. (WPs 2,4,5).

WP2: Grammar developer's tools

WP2 Requirements

Deliverables

The deliverables for WP2 are the following. As noted in the section about WP3, those two WPs are closely related, and they should be developed hand in hand.

Del. no	Del. title	Nature	Date
D 2.1	GF Grammar Compiler API	P	M12
D 2.2	Grammar IDE	P	M18
D 2.3	Grammar tool manual and best practices	RP, Main	M24

Objectives

Here we try to make a bit clearer what the functionalities of the WP2 tools are, and how they relate to the translator's tool.

We surmise that the grammar compiler's IDE is meant primarily for grammarian/engineer roles, i.e. for extending the system to new domains and languages. But it may contain facilities or components which are also relevant for the translation tool. In many scenarios, we must allow the translator to extend the system, i.e. switch to some of the last four roles. **Just how the translation tool is linked to the grammar IDE needs specifying.**

What the average user can do to fix the translation depends on how user friendly we can get. Minimally, a translator only supplies a missing translation on the fly, and all necessary adaptation is handled by the system. Maximally, an ontology or grammar needs extending as a separate chore by hand, using the grammar IDE.

An author/editor/translator can be expected to translate with the given lingware. The next level of involvement is extending the translation. This may cause entries or rules to be added to a text, company, or domain specific ontology/lexicon/grammar. If the tool is used in an organization, roles may be distributed to different people and questions of division of labor and quality control (as addressed in TF) already arise.

For it is not only, even in the first place, a question of being able to change the grammar technically, but managing the changes. A change in the source may cause improvement in some languages, deterioration in others. The author can't possibly check the repercussions in all languages. Assume each user site makes its own local changes. How many different versions of MOLTO lingware will there be? One for each website maintained with MOLTO? – how can sites share problems and solutions? A picture of a MOLTO community not unlike the one envisaged for multilingual ontology management

TF starts to form. The challenge is analogous to ontology evolution. There are hundreds of small university ontologies in Swoogle. Quality can be created in the crowd, but there must be an organisation for it (cf. Wikipedia).

The MOLTO worfklow and role play must be spelled out in the grammar tool manual (D 2.3) and the MOLTO translation tools / workflow manual (D 3.3). **We should start writing these manuals now, to fix and share our ideas about the user interfaces.**

The way disambiguation now works is that translation of a vague source against a finer grained target generates the alternative translations with disambiguating metatext to help choose the intended meaning; for example, when the user types "I love you" in Phrasebook, the program gives all 8 combinations of "I ({female, male}) love you ({polite, familiar}; {female,male})". Compare to (Boitet et al. 1994) dialogue based MT system Lidia. This facility could link to the ontology as a source of disambiguating metatext, either from meta comments or directly verbalised from ontology.

Some of the GF 3.2 features, like parse ranking and example based grammar generation, have consequences to front end design, as enabling technology.

WP2 evaluation

The role requirements for extending the system remain quite high, not because of the requirements on the individual skills, but because it is less common to find their combination in one person. The user requirements entail an important evaluation criterion: the guidance provided by MOLTO. It should also lead to system requirements, like online help, examples, profiling capabilities.

One part of MOLTO adaptivity is meant to come from the grammar IDE. Another part should come from ontologies. While the former helps extending GF "internally", the latter should allow bringing in semantics and vocabulary from OWL ontologies. We discuss these two parts in this order.

`[DoW 8 Grammar engineering for new languages]`

> *In the MOLTO project, grammar engineering in GF will be further improved in two ways:*
> *• An IDE (Integrated Development Environment), helping programmers to use the RGL and manage large projects.*
> *• Example-Based Grammar Writing, making it possible to bootstrap a grammar from a set of example translations.*
> *The former tool is a standard component of any library-based software engineering methodology. The latter technique uses the large-coverage RGL for parsing translation examples, which leads to translation rule suggestions.*

The task of building a new language resource from scratch currently is described in (Ranta 2010). As this is largely a one-shot language engineering task outside of MOLTO (MOLTO was supposed to have its basic lingware done ahead of time), it should not call for evaluation here.

Building a multilingual application for a given abstract domain grammar by way of applying and extending concrete resource grammars can use a lighter process. The proposed **example-based grammar writing process** is described in the Phrasebook deliverable 10.2 (Caprotti et al. 2010). The tentative conclusions were:

`[D10.2 MOLTO web service, first version]`

> • The grammarian need not be a native speaker of the language. For many languages, the grammarian need not even know the language, native informants are enough. However, evaluation by native speakers is necessary.
> • Correct and idiomatic translations are possible.

- A typical development time was 2-3 person working days per language.
- Google translate helps in bootstrapping grammars, but must be checked. In particular, we found it unreliable for morphologically rich languages.
- Resource grammars should give some more support e.g. higher-level access to constructions like negative expressions and large-scale morphological lexica.

Effort and cost for each language are estimated in (Caprotti et al. 2010). The phrasebook deliverable is one simple example what can be done to evaluate the grammar work package's promises. The results from the Phrasebook experiment may be positively biased because the test subjects were very well qualified. But this and similar tests can be repeated with more “ordinary people”, and changes in the figures followed as the grammar IDE is developing.

It could be instructive to repeat the exact same test with different subjects and compare the solutions, to see how much creativity was involved in the solutions. The less there is variation the better the chances to automate the process. Even failing that, analysis of the variant solutions could help suggest guidelines and best practices to the manual. Possible variation here also raises the issue of managing changes in a community of users.

WP5: Statistical and robust translation

WP5 Requirements

Deliverables

Del. no	Del. title	Nature	Date
D 5.1	Description of the final collection of corpora	RP	M18
D 5.2	Description and evaluation of the combination prototypes	RP	M24
D 5.3	WP5 final report: statistical and robust MT	RP,Main	M30

Objectives

```
[DoW  10 Robust and statistical translation methods]
```

The concrete objectives in this proposal around robust and statistical MT are:

*• **Extend the grammar-based approach** by introducing probabilistic information and confidence scored predictions.*
*• **Construct a GF domain grammar** and a domain-adapted state-of-the-art SMT system for the Patents use case.*
*• **Develop combination schemes** to integrate grammar-based and statistical MT systems in a hybrid approach.*
*• Fulfill the previous objectives on a variety of language pairs of the project (**covering three languages at least**).*

Most of the objectives depend on the Patents corpus. Even the languages of study depend on the data that the new partner provides. In order to compensate the delay due to this both in WP5 and mainly in WP7 we started working here on hybrid approaches. The methodology now is to develop hybrid methods in a way independent of the domain and data sets used, so that they can be later adapted to patents.

We already have the European Parliament corpus compiled and annotated for English and Spanish. Languages will probably finally be English, German, and Spanish or French, so as soon as this is confirmed the final general-purpose corpus can be easily compiled. The depth of the annotation will depend on the concrete languages and the available linguistic processors.

The DoW describes the hybrid MT systems we consider to include. The baseline is clear. In fact, one can define three baselines: a raw GF system, a raw SMT system and the naïve combination of both. Regarding real hybrid systems there is much more to explore. Here we list four approaches to be pursued:

- Hard integration. Force fixed GF translations within a SMT system.
- Soft integration I. Led by SMT. GF partial output, as phrase pairs, is integrated as a discriminative probability feature model in a phrase-based SMT system.
- Soft integration II. Led by SMT. GF partial output, as tree fragment pairs, is integrated as a discriminative probability model in a syntax-based SMT system.

• Soft integration III. Led by GF. Complement with SMT options the GF translation structure and perform statistical search to find the final translation.

At the moment, we are able to obtain phrases and alignments from a GF-generated synthetic corpus. This is a first step for the hard integration of both paradigms, and also for the soft integration methods led by SMT. We are currently going deeper into the latter, as it is a domain independent study. In the evaluation process, these families of methods will be compared to the baseline(s) introduced above according to several automatic metrics.

WP5 evaluation

WP5 is going to have its own internal evaluation complementary to that of WP9. Since statistical methods need of fast and frequent evaluations, most of the evaluation within the package will be automatic. For that, one needs to define the corpora and the set of automatic metrics to work with.

Corpora

Statistical methods are linked to patents data. This is the quasi-open domain where the hybridization is going to be tested. The languages of the corpus are not still completely defined, but by looking at other works with patents we guess they will probably be English, German, and French or Spanish.

Besides the large training corpus, we need at least two smaller data sets, one for development purposes and another one for testing. The order of magnitude of these sets is usually around 1,000 aligned segments or sentences. We expect to reach this size, but the final amount will depend on the available data.

Metrics

BLEU (Papineni et al. 2002) is the de facto metric used in most machine translation evaluation. We plan to use it together with other lexical metrics such as WER or NIST in the development process of the statistical and hybrid systems.

Lexical metrics have the advantage of being language-independent, since most of them are based on n-gram matching. However, they are not able to catch all the aspects of a language and they have been shown not to always correlate well with human judgments. So, whenever it is possible, it is a good practice to include syntactic and/or semantic metrics as well.

The IQmt package provides tools for (S)MT translation quality evaluation. For a few languages, it provides metrics to do this deep analysis. At the moment, the package supports English and Spanish, but other languages are planed to be included soon. We will use IQmt for our evaluation on the supported language pairs.

WP4 Knowledge engineering

Ontotext contributions to MOLTO through WP4 are

• Semantic infrastructure

• Ontology-grammar interoperability

WP4 requirements

Deliverables

Del. no	Del. title	Nature	Date
D 4.1	Knowledge Representation Infrastructure	RP	M8
D 4.2	Data Models, Alignment Methodology, Tools and Documentation	RP	M14
D 4.3	Grammar-Ontology Interoperability	P,Main	M18

Knowledge Representation Infrastructure

From Ontotext webpages, we can guess that the infrastructure builds on the following technologies:

•KIM is a platform for semantic annotation, search, and analysis

•OWLIM is the most scalable RDF database with OWL inference

•PROTON is a top ontology developed by Ontotext.

Milestone MS2 says the knowledge representation infrastructure is opened for retrieval access to partners at M6. The infrastructure deliverable D4.1 is due at M8.

Grammar-ontology interoperability

```
[DoW  7 Grammar-ontology interoperability for translation and retrieval]
```

> *At the time of the TALK project, an emerging topic was the derivation of dialogue system grammars from OWL ontologies. A prototype tool for extracting GF abstract syntax modules from OWL ontologies was thereby built by Peter Ljunglöf at UGOT. This tool was implemented as a plug-in to the Protégé system for building OWL ontologies and intended to help programmers with OWL background to build GF grammars. Even though this tool remained as a prototype within the TALK project, it can be seen as a proof of concept for the more mature tools to be built in the MOLTO project.*

A direct way to map between ontologies and GF abstract grammars is a mapping between OWL and GF syntaxes.

> *In slightly simplified terms, the OWL-to-GF mapping translates OWL's classes to GF's categories and OWL's properties to GF's functions that return propositions. As a running example in this and the next section, we will use the class of integers and the two-place property of being divisible ("x is divisible by y"). The correspondences are*

as follows:

```
Class(pp:integer ...)  <==>      cat integer ;

ObjectProperty(pp:div  <==>      fun div :
  domain(pp:integer)             integer -> integer -> prop ;
  range(pp:integer))
```

Less syntax-directed mappings may be more useful, depending on what information is relevant to pass between the two formalisms. The mapping is then also less generic, as it depends on the intended use and interpretation of the ontology. The mapping through SPARQL queries below is one example. A mapping over TF could be another one.

Note also that the OWL to GF mapping also allows a wider human input to GF. OWL ontologies are written by humans (at present at least, by many more humans than GF grammars).

MOLTO website gives detail what is going to delivered first by way of ontology-GF interoperability. The first round uses GF grammar to translate NL questions to SPARQL query language ("Clarification of GF<->Ontology interoperability"). The ontology-GF mapping here is a NL interface to PROTON ontologies, by way of parsing (fixed) NL to (fixed) GF trees and transforming the trees into SPARQL queries to run on the ontology DB.

Indirectly, this does define a mapping between (certain) GF trees and RDF models, using SPARQL in the middle. SPARQL is not RDF but a SPARQL query does retrieve a RDF model given a dataset, but the model depends on the dataset. With an OWL reasoner thrown in, we can get OWL query results.

What WP3 had in mind is a tool to translate between OWL models and GF grammars, i.e. convert OWL ontology content into GF abstract syntax. This tool is forthcoming next, which was confirmed by email from Petar.

The translation tools WP3 will consider using TermFactory multilingual ontology model and tools as middleware between (non-linguistic) ontology and GF grammar. The idea is to (semi)automatically match or bridge third party ontologies to TF, a platform for collaborative development of ontology-based multilingual terminology. It then remains to define an automatic conversion between TF and GF.

The Varna meeting should adjudicate between WP3 and WP4 here.

A concrete subtask that arises here is to define an interface between the knowledge representation infrastructure (due Nov 2010) and TF (finished in ContentFactory project end of 2010).

WP4 evaluation

Since the aims are more related to use cases and framework development, than enhancing performance of existing technologies, the evaluation to be done during the project will be more of a qualitative than quantitative kind.

The evaluation of these features should reflect and demonstrate the multiple possibilities of GF that are gained through inter-operation with external ontologies. The evaluation of progress will exploit proof-of-concept demos and plans for further development. For further discussion, see ("Evaluation of Ontology Features in MOLTO").

References

Boitet, Christian et al. 1994. Multilingual Dialogue-Based MT for monolingual authors: the LIDIA project and a first mockup. *Machine Translation*, Volume 9, Number 2, p 99-132. URL http://www.springerlink.com/content/kn8029t181090028/.

Caprotti, Olga et al. 2010. *D10.2 MOLTO web service, first version*. http://www.molto-project.eu/sites/default/files/D10.2.pdf.

Drupal-Translation. "Drupal Translation Management". URL http://drupal-translation.com/content/drupal-translation-management.

Koponen, Maarit. 2010. "Metric Comparison". URL https://kitwiki.csc.fi/twiki/pub/MOLTO/EvaluationCookbook/metric_comparison.ods.

Koponen, Maarit 2010 (forthcoming). Assessing Machine Translation Quality with Error Analysis. MikaEL – Electronic proceedings of the KäTu symposium on translation and interpreting studies, Vol. 4 [online]. Available at: http://www.sktl.net/mikael/

Leusch, Gregor. Jul 2005. *Evaluation Measures in Machine Translation*. URL http://www-i6.informatik.rwth-aachen.de/publications/download/341/Leusch—2005.pdf.

MOLTO. "Clarification of GF<->Ontology interoperability". URL http://www.molto-project.eu/node/987.

MOLTO. "Deliverables". URL http://www.molto-project.eu/workplan/deliverables.

MOLTO. "WP6: Case Study Mathematics". URL http://www.molto-project.eu/node/859.

MOLTO. "WP9: User Requirements and Evaluation". URL http://www.molto-project.eu/node/865.

Ranta, Aarne. 17 May 2010. *Creating Linguistic Resources with the Grammatical Framework*. LREC. URL http://www.grammaticalframework.org/doc/gf-lrec-2010.pdf.

Schonfeld, Erick. 29 Sep 2009. "Facebook Spreads Its Crowdsourced Translations Across the Web, And The World". *TechCrunch*. URL http://techcrunch.com/2009/09/29/facebook-spreads-its-crowdsourced-translations-across-the-web-and-the-world.

UHEL Evaluation Cookbook. "E-mail from Petar". https://kitwiki.csc.fi/twiki/bin/view/MOLTO/MoltoOntologyEvaluationPlanWP4.

UHEL Evaluation Cookbook. "Evaluation of Ontology Features in MOLTO". URL https://kitwiki.csc.fi/twiki/bin/view/MOLTO/MoltoOntologyEvaluationPlanD91.

UHEL Evaluation Cookbook. "The cost of human reference translation". URL https://kitwiki.csc.fi/twiki/bin/view/MOLTO/EvaluationCookbook?skin=clean.nat,editdefault,pattern#The_cost_of_human_reference_tran.