Blaise Translation Challenges: Versioning, Multimode and Exporting

M.G.J. Martens MSc, CentERdata

Abstract

CentERdata has developed a system for managing translations for CAPI Blaise questionnaires for several international panel studies. This was done with an online tool called the Language Management Utility (LMU). It is highly configurable and support includes various element types, assignments, workflows, states, versions, users, modules and languages.

Recently the LMU has been extended. The backend was redesigned to provide new functionality. Rather than fixed versions, every translation is saved, which makes it possible to revert and fully analyse the translation process. Several methods of CAWI integration were tested and implemented. Blaise questionnaires can now be uploaded and changes since the last upload are detected and automatically updated and flagged in LMU. Questionnaire routing is stored in the LMU.

Several new exports were added: each type of questionnaire can be reviewed online in any language with texts imported from the LMU database. Multilingual Blaise source code can be generated from the LMU directly. It is possible to browse through the questionnaire in all stages of development in track changes mode. Several new exports in Excel are available. Questionnaires can be exported in DDI3-compatible format.

The new LMU features provide the means to support a larger part of the questionnaire design process. They can be regarded as the first steps towards an online Questionnaire Management Environment for international studies.

1 Introduction

Since 2002 CentERdata has been developing a system to guide the translation process for multilingual Blaise questionnaires. Initially developed for the SHARE project. **The Survey of Health, Ageing and Retirement in Europe (SHARE) is a multidisciplinary and cross-national panel database of micro data on health, socio-economic status and social and family networks of more than 55,000 individuals from 20 European countries aged 50 or over.** After a few waves of SHARE it was fine-tuned and also opened up to other parties

This paper discusses some projects we started but not yet finished, some inspiration we picked up along the way and how this all cumulated in a complete redesign of the Language Management Utility (LMU).

Originally this system consisted of a (web-) frontend and a backend. The frontend allowed translators to add their translation through a website, LMU. Texts from a structured Blaise questionnaire were manually copied over into the LMU. After an iterative process of translating, redesigning, testing this delivered us a final version that could be used in the field. On the backend two tools were implemented to manage these translated texts;

- A 'Blaise Generator' developed in Visual Basic, that pasted translated texts into a generic version,
- A 'Paperversion Generator' that exported a textual overview of the questionnaire and routing.

This process and the tools involved are described more in depth in the 2009 IBUC paper "Managing Translations for Blaise Questionnaires" (*Martens, et al, 2009*).

In following waves of SHARE the LMU expanded in many dimensions. The functionality of the web-frontend grew; it was also made possible to translate texts for the Share Sample Management Systems, help files and an implementation of the Event History Calendar.

For the Understanding Society study a management layer was build around the LMU to better manage the translation process. Problems were encountered with the Blaise DEP not supporting Unicode and a Unicode Enabled data entry program Unitip was created. This tool and process are described in the 2009 IBUC paper "An End-to-End Solution for Using Unicode with Blaise to Support Any Language" (*Amin et al., 2009*).

Early 2011 CentERdata was asked to develop and host a multilingual web questionnaire for a dozen European countries. Although the LMU tools were originally designed for CAPI questionnaires only, it was decided to try to adapt the code to get it working for web questionnaires as well. The experiences of rebuilding the LMU for this project could be used to design a new system that would fully support multi-mode. In section 2 our findings and some ideas we got in the process are described.

Over the years the LMU database got filled with question texts, answers, fills, labels and their translations in various stages of the questionnaire design over multiple studies over multiple waves in over 40 languages with various character sets. We decided to see if it was feasible to build alternative interfaces on this data; interfaces to analyze the translation process and questionnaire development or a system that could function as a question bank. To see what exports could be made. Maybe even create new questionnaires directly from the questions in the database. In section 3 some experiments with new interfaces on the data, some ideas on exports and linkage to a data dissemination system is discussed.

Based on the findings from rebuilding the LMU for web mode and the discussion and experiments that were done for question banks some architectural flaws were discovered. The code base and database design needed to be completely rewritten to be ready for challenges in the near future. In section 4 the new design and interface will be discussed.

2 Translating for web mode

For a study on sustainability CentERdata was asked to create a web questionnaire for 12 European countries. (Austria, Denmark, Finland, France, Germany, Hungary, Italy, Netherlands, Poland, Spain, Sweden, UK). It would seem logical to use the LMU to accommodate the translation process. But it was originally built to support the translation process for Blaise Questionnaires in CAPI mode only. This meant the Blaise source code was structured in a manner that could easily communicate with the LMU. For example the assumption was made a question existed of the following 6 components, making it easier to parse the source code:

```
QuestionName (QuestionTag)
 "QuestionText
InterviewerInstruction"
 /"Description":
Answer
```

Other texts were loaded using fills, either attached to a question or defined on a global level. One of the key thoughts behind the LMU is that translators should be presented the questions in the order and with the texts of the questionnaire that will be available during the field. A translator should translate questions not text strings without context.

In our experience when programming for CAWI the structure of the source code can become quite different than for CAPI. For our CAPI environments we normally show 1 question on screen. In CAWI more tables are used; more questions are on the screen simultaneous, the layout tends to have impact on the way the questionnaire is programmed. Therefore the strict definition of a question we used in LMU earlier didn't fit this mode. Several approaches were tried to adapt the LMU to support more complex structured questionnaires.

First we attempted to extend the parsing of the source code. We tried to catch tables, blocks and more difficult structures and stored them in the databases and displayed them in the LMU web environment. These more complex structures were stored in xml structures in the original LMU tables. This approach felt a bit stupid. Trying to parse source code looking for structures seemed an effort that would only set new boundaries and would again force us to limit the ways in which we could structure our Blaise programs.

It was decided to try the reverse approach; instead of trying to get the structured translatable elements out of a Blaise questionnaire, we would identify the translatable elements in the Blaise source code. We coded question texts with 'tags'. To the text strings in the original source code some xml like tags (</>) surrounded the original text. This approach would also allow using the LMU for other purposes as well, for other structured texts we could design tags that could be understood and displayed by the LMU

```
eca132
"<t id='IS:eca132' type='questionnaire:text'>Do you want to receive
a feedback report that compares your response to the survey with the
average response of enterprises in your sector and country?
The report will be send to the e-mail address on which you received
the invitation to complete this survey.</t>": T_YNN
```

These tagged strings made it possible to easily import the source code into the LMU, extract the tagged elements and store in the proper place based on their type and id. Also returning a translated text back into the original questionnaire was easy. Simply search the original tags and replace the full string with its translation.

```
eca132
"@#Haluatteko palauteraportin, jossa vertaillaan antamianne
vastauksia maanne ja liiketoiminta-alanne keskiarvoihin?
Raportti l&#228;hetet&#228;&#228;n samaan
s&#228;hk&#246;postiosoitteeseen, johon saitte kutsun
t&#228;ytt&#228;&#228; t&#228;m&#228; kysely.@#": TYesNo
```

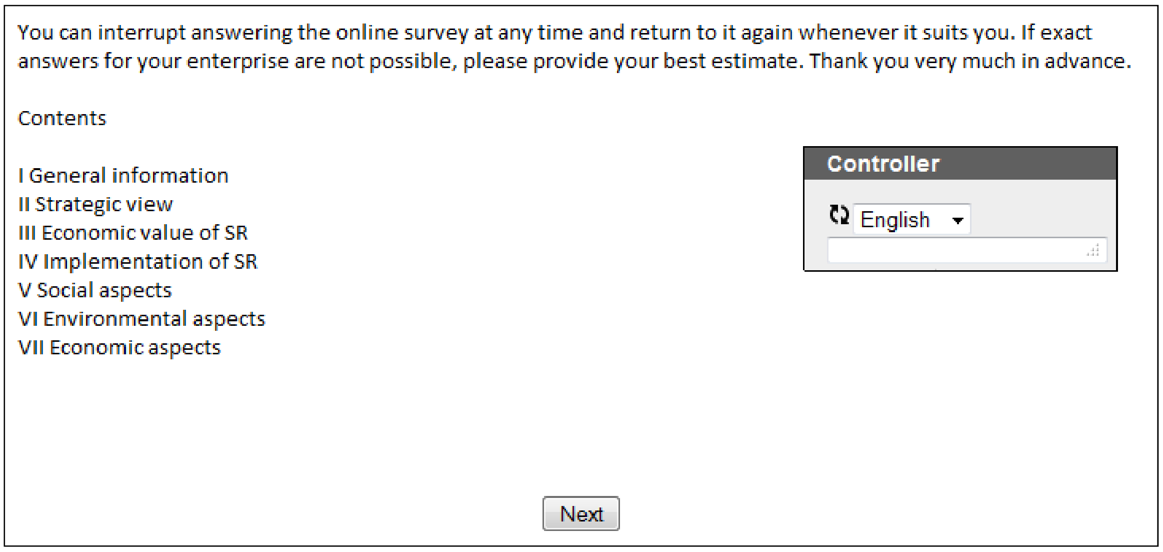
In a web questionnaire html codes for problem characters or Unicode can be used as well.

This worked reasonably well. It was annoying to manually tag the Blaise source, but you had to do it only once. Our questionnaire scripters got used to it fairly quick.

With this structure in place and the twelve countries responding to the questionnaire it was time to see if we could use this in a more generic manner. A special version of the questionnaire was exported. It didn't replace translation texts in the source code but replaced the tags with set a html-div block with an id and class.

```
eca132
"@#<div class=""RemoteTextLoad"" id=""IS:eca132""></div>@#": TYesNo
```

Using some JavaScript, adapting the style sheet and introducing floating div control menu this questionnaire was tricked into loading questionnaires directly from the LMU database, extracting translated texts at runtime. This seemed to give us the possibility to set up a version of the questionnaire where translation changes could be reviewed immediately without running a process to paste translations into source code and compile it. This meant testing of the questionnaire translation could be done immediately. This could shorten development time.



You can interrupt answering the online survey at any time and return to it again whenever it suits you. If exact answers for your enterprise are not possible, please provide your best estimate. Thank you very much in advance.

Contents

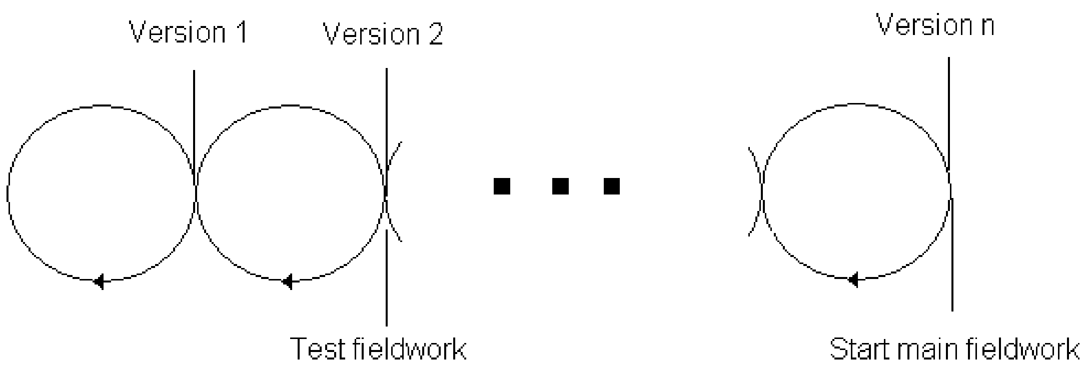
I General information
II Strategic view
III Economic value of SR
IV Implementation of SR
V Social aspects
VI Environmental aspects
VII Economic aspects

Unfortunately inserting the tagged-elements in the generic source code was not very user-friendly. To a certain extend we could have automated tagging elements but we would be trapped again in parsing source code.

This let us to investigate whether it is possible not to use the source code at all for this purpose. The backend of our translation software and Unitip were already built on the Blaise API. So it wasn't a big step to use the Blaise API to acquire the structure directly from the compiled questionnaire. This would however mean we had to redesign the LMU's database architecture completely to be more compatible with the internal structure of the Blaise Datamodel.

3 Ideas

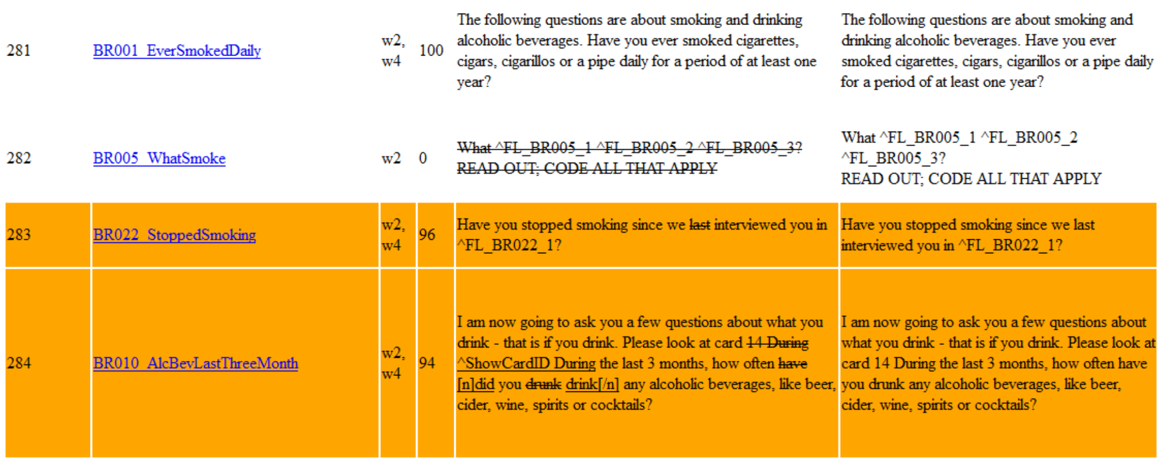
The LMU for Share contains 4 waves of questions. Each wave consists between 7 and 13 cycles in which the questionnaire was adapted or translated. Some cycles end with a version of the questionnaire that is fielded; a pilot, pretest and main field work version. Each cycle itself consists of sub cycles where the questionnaire is fine-tuned, smaller bugs are resolved or translations are improved.



In each cycle a version of the questionnaire is saved for each translation. This led to a large collection of questions, questionnaire elements and their translations.

We got the idea that analyzing the translation process might say something about the quality of the questionnaire. If a question changes a lot through the translation process, this is an indication that it is hard to translate and therefore it is likely problems will occur that will influence the data collected with these questions.

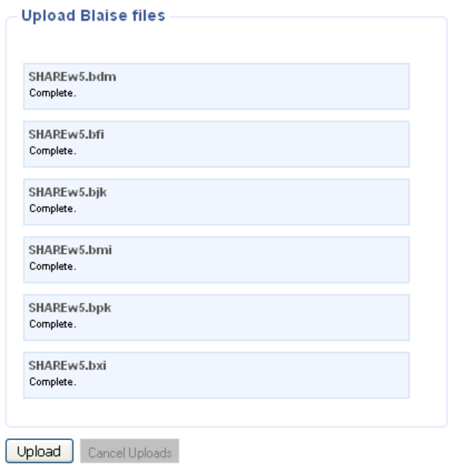
To determine what questions changed and to what extend they changed the Levenshtein distance divided by the maximum number of strings provided was used. The Levenshtein distance is the number of changes you have to make to change a string into another string. Also an interface based on a ‘track changes view’ was created. This measure can also be used to determine to what extend a translation is changed. Comparing this with other translations or with the change in the original question could also be used to track potential problems.



281	BR001_EverSmokedDaily	w2, w4	100	The following questions are about smoking and drinking alcoholic beverages. Have you ever smoked cigarettes, cigars, cigarillos or a pipe daily for a period of at least one year?	The following questions are about smoking and drinking alcoholic beverages. Have you ever smoked cigarettes, cigars, cigarillos or a pipe daily for a period of at least one year?
282	BR005_WhatSmoke	w2	0	~~What ^FL_BR005_1 ^FL_BR005_2 ^FL_BR005_3? READ OUT; CODE ALL THAT APPLY~~	What ^FL_BR005_1 ^FL_BR005_2 ^FL_BR005_3? READ OUT; CODE ALL THAT APPLY
283	BR022_StoppedSmoking	w2, w4	96	Have you stopped smoking since we ~~last~~ interviewed you in ^FL_BR022_1?	Have you stopped smoking since we last interviewed you in ^FL_BR022_1?
284	BR010_AlcBevLastThreeMonth	w2, w4	94	I am now going to ask you a few questions about what you drink - that is if you drink. Please look at card ~~14 During~~ ^ShowCardID During the last 3 months, how often ~~have~~ [n]did you ~~drunk~~ drink[/n] any alcoholic beverages, like beer, cider, wine, spirits or cocktails?	I am now going to ask you a few questions about what you drink - that is if you drink. Please look at card 14 During the last 3 months, how often have you drunk any alcoholic beverages, like beer, cider, wine, spirits or cocktails?

To properly analyze the translation process we however need to know all the steps a translator went through. The version based setup of the LMU was not suitable for this.

While experimenting with an automatic import of previous translations by calling the Blaise API via PHP we created a few scripts. The old VB6 Paper version generator we used to create readable overview of the questionnaire routing was rewritten in PHP, an upload form was added and this made it possible to create the paper versions online.



Upload Blaise files

SHAREw5.bdm
Complete.

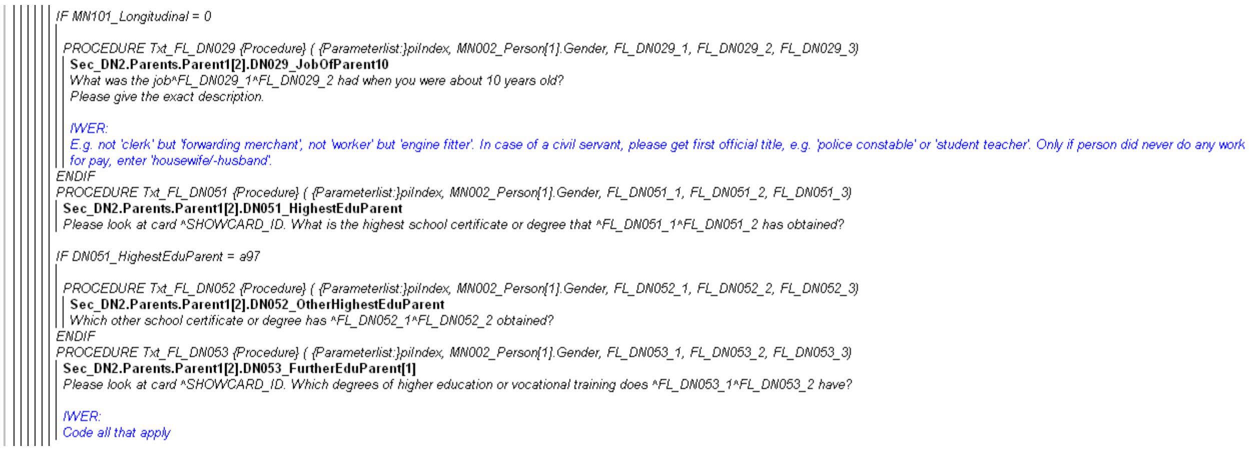
SHAREw5.bfi
Complete.

SHAREw5.bjk
Complete.

SHAREw5.bmi
Complete.

SHAREw5.bpk
Complete.

SHAREw5.bxi
Complete.



```
IF MN101_Longitudinal = 0

 PROCEDURE Txt_FL_DN029 {Procedure} ( {Parameterlist:}piIndex, MN002_Person[1].Gender, FL_DN029_1, FL_DN029_2, FL_DN029_3)
  Sec_DN2.Parents.Parent1[2].DN029_JobOfParent10
  What was the job^FL_DN029_1^FL_DN029_2 had when you were about 10 years old?
  Please give the exact description.

  IWER:
  E.g. not 'clerk' but 'forwarding merchant', not 'worker' but 'engine fitter'. In case of a civil servant, please get first official title, e.g. 'police constable' or 'student teacher'. Only if person did never do any work
  for pay, enter 'housewife/-husband'.
ENDIF
PROCEDURE Txt_FL_DN051 {Procedure} ( {Parameterlist:}piIndex, MN002_Person[1].Gender, FL_DN051_1, FL_DN051_2, FL_DN051_3)
 Sec_DN2.Parents.Parent1[2].DN051_HighestEduParent
 Please look at card ^SHOWCARD_ID. What is the highest school certificate or degree that ^FL_DN051_1^FL_DN051_2 has obtained?

IF DN051_HighestEduParent = a97

 PROCEDURE Txt_FL_DN052 {Procedure} ( {Parameterlist:}piIndex, MN002_Person[1].Gender, FL_DN052_1, FL_DN052_2, FL_DN052_3)
  Sec_DN2.Parents.Parent1[2].DN052_OtherHighestEduParent
  Which other school certificate or degree has ^FL_DN052_1^FL_DN052_2 obtained?
ENDIF
PROCEDURE Txt_FL_DN053 {Procedure} ( {Parameterlist:}piIndex, MN002_Person[1].Gender, FL_DN053_1, FL_DN053_2, FL_DN053_3)
 Sec_DN2.Parents.Parent1[2].DN053_FurtherEduParent[1]
 Please look at card ^SHOWCARD_ID. Which degrees of higher education or vocational training does ^FL_DN053_1^FL_DN053_2 have?

 IWER:
 Code all that apply
```

Other exports include overviews of fields and properties and of conditions that should hold for a field in excel. It should also be quite easy to export in xml. These are potential building blocks for data dissemination systems, question banks and the Language Management Utility.

4 Complete redesign

The experiments and ideas presented in previous sections made it clear that a redesign of the LMU could open up a whole new range of possibilities. We had to improve the versioncontrol, make the number of languages dynamic, support multiple questionnaires and make the backend system part of the webenvironment.

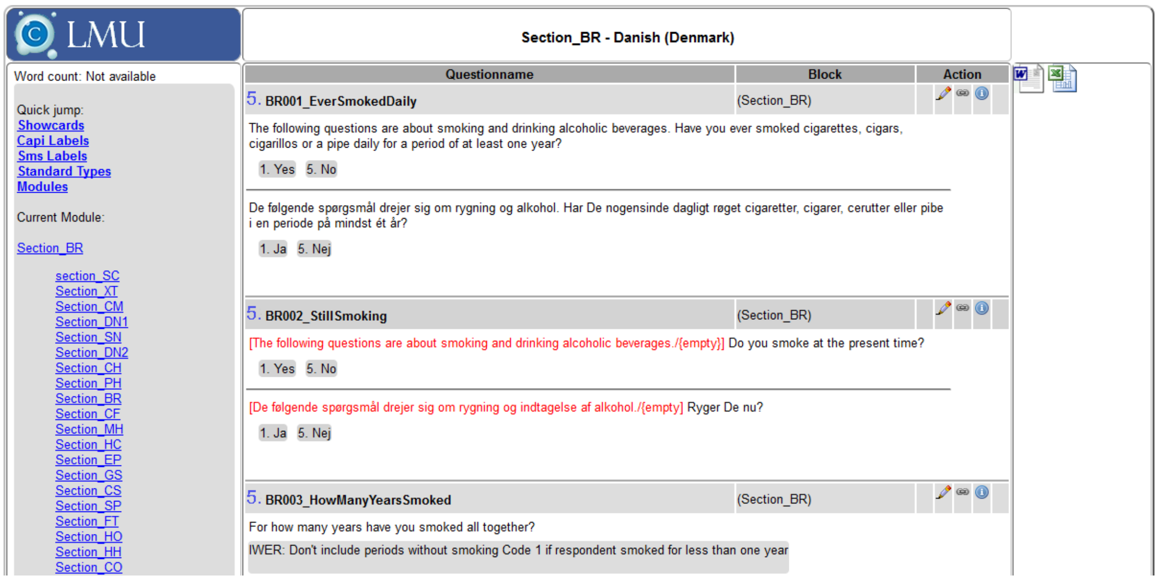
In the original design there were no dates attached to translations and there was only one save per version. This made it impossible to do proper analysis on the translation process. If every change was stored we could revert to a previous state when problems occurred. This would also open up the option to view history for each translated element to the translator. Therefore we removed the notion of versions altogether. A questionnaire gets updated, changes are now automatically tracked and flagged and a translator can come in any moment and translate elements that changed.

The translation languages were stored statically in the database before. If a language was added we had to add columns to the tables. This was adapted to a dynamic option. The elements table was introduced, it stores all translated strings with language_id as a foreign key. This was linked to a languages table that was extended to include settings for character sets for export.

The concept of procedures was unknown to the LMU before. If translations were stored in a procedure, we had to tag them with a code to find them. A procedure-table was included to find fill values that were loaded through procedures.

The LMU supported only one questionnaire per install. Now it is possible to insert multiple questionnaires simultaneous. As the number of questionnaires included grows, the questions and translations will be a rich source for new translation suggestions and make it possible to do cross questionnaires comparisons.

Translated versions can be generated directly from the web-interface. This means the Blaise Generator and Multi Blaise generator are no longer needed.


LMU

Section_BR - Danish (Denmark)

Word count: Not available

Quick jump:
Showcards
Capi Labels
Sms Labels
Standard Types
Modules

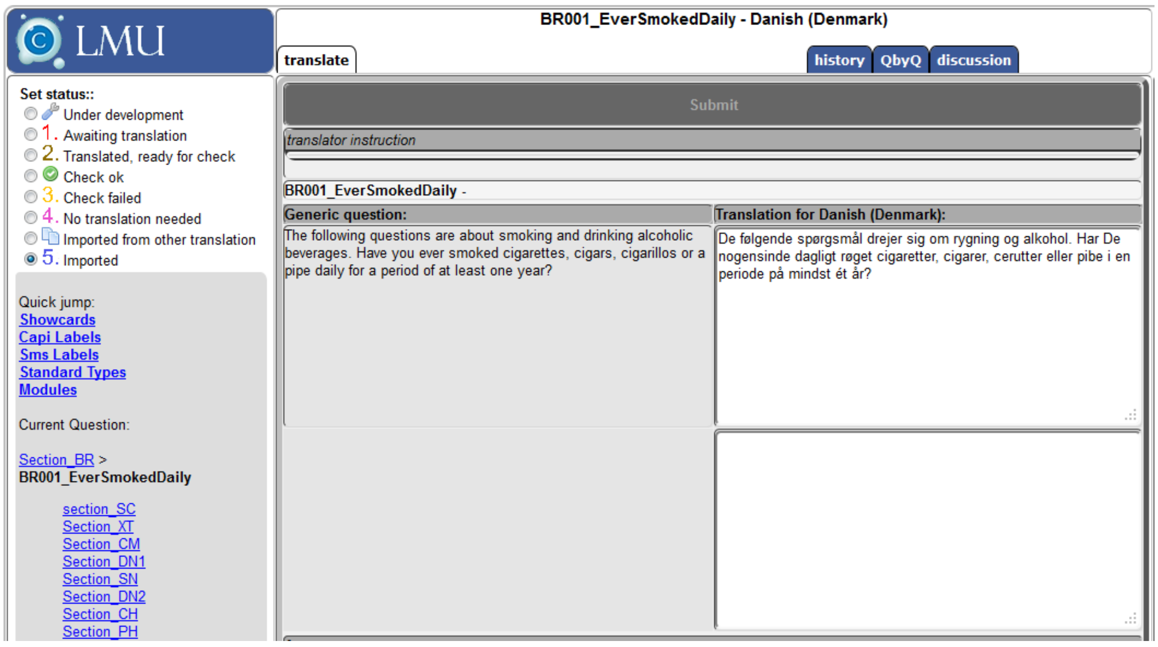
Current Module:

Section_BR

section_SC
Section_XT
Section_CM
Section_DN1
Section_SN
Section_DN2
Section_CH
Section_PH
Section_BR
Section_CF
Section_MH
Section_HC
Section_EP
Section_GS
Section_CS
Section_SP
Section_FT
Section_HO
Section_HH
Section_CO

Questionname	Block	Action
5. BR001_EverSmokedDaily	(Section_BR)	
The following questions are about smoking and drinking alcoholic beverages. Have you ever smoked cigarettes, cigars, cigarillos or a pipe daily for a period of at least one year? 1. Yes 5. No De følgende spørgsmål drejer sig om rygning og alkohol. Har De nogensinde dagligt røget cigaretter, cigarer, cerutter eller pibe i en periode på mindst ét år? 1. Ja 5. Nej		
5. BR002_StillSmoking	(Section_BR)	
[The following questions are about smoking and drinking alcoholic beverages./{empty}] Do you smoke at the present time? 1. Yes 5. No [De følgende spørgsmål drejer sig om rygning og indtagelse af alkohol./{empty}] Ryger De nu? 1. Ja 5. Nej		
5. BR003_HowManyYearsSmoked	(Section_BR)	
For how many years have you smoked all together? IWER: Don't include periods without smoking Code 1 if respondent smoked for less than one year		

On the left side of the interface, the routing options are displayed, making it easier to access other modules. In the center a tab-controlled work-area is implemented. We can add new functionality easily by adding new tabs.


LMU

BR001_EverSmokedDaily - Danish (Denmark)

translate | history | QbyQ | discussion

Set status::
- ○ Under development
- ○ 1. Awaiting translation
- ○ 2. Translated, ready for check
- ○ Check ok
- ○ 3. Check failed
- ○ 4. No translation needed
- ○ Imported from other translation
- ● 5. Imported

Quick jump:
Showcards
Capi Labels
Sms Labels
Standard Types
Modules

Current Question:

Section_BR >
BR001_EverSmokedDaily

section_SC
Section_XT
Section_CM
Section_DN1
Section_SN
Section_DN2
Section_CH
Section_PH

Submit

translator instruction

BR001_EverSmokedDaily -

Generic question:	Translation for Danish (Denmark):
The following questions are about smoking and drinking alcoholic beverages. Have you ever smoked cigarettes, cigars, cigarillos or a pipe daily for a period of at least one year?	De følgende spørgsmål drejer sig om rygning og alkohol. Har De nogensinde dagligt røget cigaretter, cigarer, cerutter eller pibe i en periode på mindst ét år?

The web-environment is a huge improvement. The translators in the fifth wave of share enjoyed working with this new approach to translating. The automatic import and direct options to export source code shortened the development process tremendously. The new Blaise DEP 4.8.3 supports different character sets much better than before, including full Unicode support. It seems we can now use the Blaise DEP to display questionnaires again and don't need our Unitip tool anymore for this purpose. This also means we don't have to work around the problems with accessing the activelanguage-property anymore. The described changes open up a wide range of future improvements.