

D 3.3 Test report SEAL Project

Gijs van den Brink (O&i)
Emile de Maat (UvA)

With contributions of:
Dutch House of Representatives
Austrian House of Representatives
Italian Chambers of Deputy
Italian Senate
Cirsfid- University of Bologna
CNR-ITTIG

Version 1.0
Datum 14-07-2008

Table of Contents

1. CONTEXT OF THE DELIVERABLE.....	3
2. APPROACH.....	4
2.1. PURPOSE.....	4
2.2. METHOD.....	4
2.3. EQUIPMENT.....	5
2.4. TEST PARTICIPANT PROFILES.....	5
3. THE SYSTEM ENVIRONMENT.....	6
4. THE TESTED PROCESS.....	7
4.1. DUTCH NATIONAL TASKS.....	7
<i>Test Tasks</i>	7
<i>Test Cases</i>	7
4.2. ITALIAN NATIONAL TASKS.....	8
<i>Test Tasks</i>	8
<i>Test Cases</i>	9
4.3. WORKFLOW.....	9
5. FINDINGS DUTCH CASE.....	10
6. FINDINGS ITALY.....	14
6.1. NORMA EDITOR.....	16
6.2. XMLLEGESEDITOR.....	18
7. GENERAL OVERVIEW.....	21
7.1. GENERAL FINDINGS.....	21
7.2. FUTURE WORK.....	22
ATTACHMENTS.....	24
APPENDIX A TEST CASES.....	25
APPENDIX B EVALUATION QUESTIONNAIRE.....	29
APPENDIX C WORKFLOW.....	30
WORKFLOW DUTCH CASE.....	30
XML DUTCH CASE.....	32
WORKFLOW ITALIAN CASE.....	33



1. Context of the deliverable

The SEAL project consisted of different stages, the development stage and the test/pilot stage. During the different stages a continuous evaluation cycle was used for evaluating and continued developing.

At the start of the project, the success criteria for the project were defined and measured (see D1.5 Initial Success Criteria). In addition, the users' requirements for a working environment were inventoried (D2.1 Requirements). During the development stage, the working environments were modified to comply more closely to those requirements and criteria.

During the test/pilot stage, the environments were further improved. Rapid Application Development (RAD) was used, the whole creation process consisted of multiple small cycles of building, testing and refining the drafter environment. Users were asked to give their feedback on the improvements made, and based on their feedback, further modifications were implemented. The test/pilot stage ended with a formal evaluation of the last version, in which the impact of the working environments on success criteria was measured.

This report is based on the experiences of the SEAL pilot phase from December 2007 until July 2008. In chapter 2, the test approach will be explained. After this the test cases, the tested environments will be described in chapter 3. Chapter 4 gives the test cases and test tasks for both pilot sites

The results of the test during the pilots will be given in chapter 5 and 6, in which chapter 5 describes the results of test location in the Netherlands and chapter 6 Italy. After describing the test result the workflow component will be evaluated, followed by a general conclusion in chapter 7.

The test phase took place on three different test sites, with three different tools. To make a comparison between the 'different' tools a quantitative comparison has to be made. This comparison will be analyzed in the Evaluation report. The questionnaire used can be found in Appendix B.



2. Approach

The pilot phase of the SEAL project started in December 2007. The period until June 2008 was used for testing, developing and evaluating the different editors at the different locations. The test locations were:

- Dutch House of Representatives
- Austrian House of Representatives
- Italian Chamber of Deputies
- Italian Senate of Republic

The alternative editors are:

- xmLegesEditor (ITTIG).
- The Norma editor (CIRSFID).
- The Metalex General Purpose Editor (GPE).

As each test location has its own local drafting processes and language, it was impossible to test each editor at each location. This would require modifications that were not possible within the scope of this project.

With testing three different editors it is possible to create synergy in developing and implementation of the working environments. Sharing lessons learned will give suggestions for implementing an Europe-wide working environment.

2.1. Purpose

The primary purpose of the testing phase was to test the functionalities of the editor and find out if these functionalities match with the requirements the end-user defined when determining the initial success criteria. A secondary purpose was test the usability of the working environment. The test performed where on technical, user and acceptance test level.

As Rapid Application Development (RAD) was used, the whole creation en test process consisted of multiple small cycles of building, testing and refining the drafter environment. (See also D2.1 Requirements).

The outcomes of the test cycles were input for development. This report gives the most important findings and gives the solutions that were created.

2.2. Method

The working environments were tested with legal drafters from different parliaments. For the Dutch case, the tests were hosted at the Parliament in The Hague, with support from O&I. The Italian tests were hosted by the Senate of the Republic and the Chamber of Deputies. The Austrian tests were hosted by the Leibniz Center for Law.

Besides technical testing at the pilot locations, the SEAL test phase consisted of three different steps:

1. Interviews During and after the test the participants were interviewed about the working environment and their experiences with it. Also the participants were asked about recommendations for



further developing and implementation of the working environment.

2. Solving test tasks During the different test periods there were tests consisting of tasks that are part of the daily work of the participants. A detailed description is given in chapter 4.
3. Debriefing After the test period the participants were asked to fill in a questionnaire about their experiences. This questionnaire was based on the initial questionnaire that was used to measure the existing situation and defining requirements for development.

The test phase was an iterative process. During this process different pilot sessions were held. Based on the findings of these sessions, the editors and working environments were adjusted to fix problems and get user acceptance.

2.3. Equipment

The tests were performed on different laptops and the workstations of the test participants. On the workstations, in the working environment (in the Dutch case the integrated version (MetaVex/Be Informed Studio) was installed.

The Italian pilot tested the integrated versions of xmLegesEditor and Norma-Editor with Be Informed, although the two Italian Houses had a pre-existing propriety information system for the workflow (one for each house)¹.

In the Italian Parliament the "xml-Leges" project is started, that aims to the interchange document flow of the official version of the drafting-law during the law-making process using XML, certificated e-mail, peer-to-peer architecture. For this reason the Be Informed platform was considered marginally due to the presence of this new innovative system. The Be Informed environment was not used during testing in Italy.

2.4. Test Participant Profiles

The tests were carried out by test participants with the following attributes:

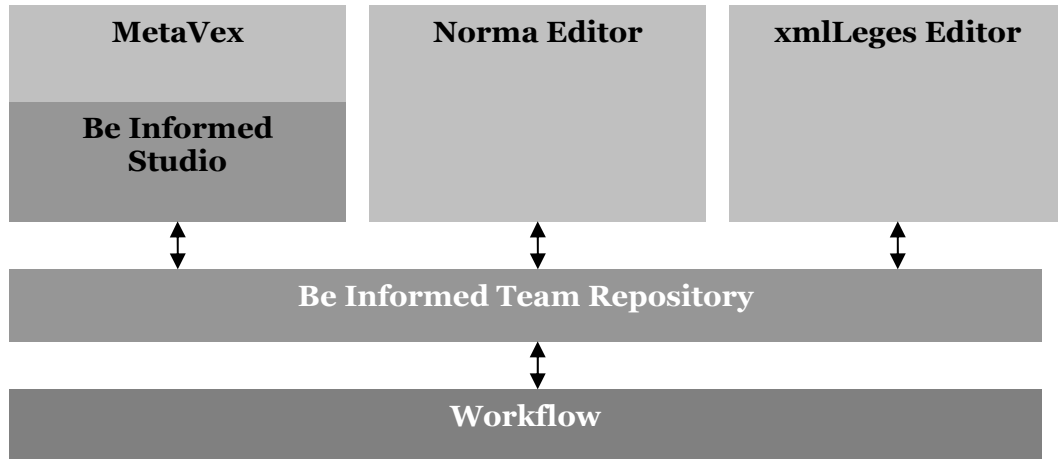
- Involved in legislative text writing on daily basis
- Experiences with the current editors used for legislative text writing. Experience differs from 1 till 20 years experience with legislative text writing.
- Both male and female testers.

¹ While in all three cases the editors were integrated with BeInformed, the manner of integration varied. The MetaVex editor was fully integrated at interface level with the BeInformed product suite, while Notma Editor and xmLeges were integrated through the BeInformed Team Repository.



3. The system environment

A short description of the tested system environment shows the dependency between the different tools.



The workbench will be sustained by Be Informed, who will exploit it as well (and thus charge any end users). Be Informed will act as the system integrator for the add-ons and editors and will make sure that the working environment will be offered in one single package. The user-organizations that are pilot site in this project may use the pilot sites environment that results of this project one year after the projects ending free of charge. After that period these organizations are entitled to a discount of 10 percent of the official license fee.

Use of the xmLeges Editor is free of charge among Public Administrations and others. This editor is released according to a GPL license implying that it can be used and updated within the limits of GPL license. It is specialized in supporting the legal drafting phases; start from scratch or for readjusting an existing text.

The Norma Editor is released under proprietary license "like-GPL": The source code is released, it is modifiable, it is possible to copy, to use and to distribute it, but it is not possible to derivate commercial products without a previous agreement with the CIRSFID. The Norma-Editor is specialized in supporting the legal drafting during the consolidation phase (amendments to a draft proposal).

The Metalex General Purpose Editor (GPE) is an open source editor. It is based on Vex, which is released under the GNU Lesser General Public License (LGPL), meaning that it will always be free to use, modify, and re-distribute within the limits of the LGPL license and also specialized in supporting the legal drafting during the consolidation phase (amendments to a draft proposal).



4. The tested process

During the test meetings, a set of tasks was performed. Besides performing the different tasks, the usability and technical functionalities of the editor were tested.

4.1. Dutch National Tasks

All tasks have been performed on bill 30411 (Regels omtrent instanties die verantwoordelijk zijn voor handhaving van de wetgeving inzake consumentenbescherming (Wet Handhaving consumentenbescherming)). The test participants got the assignment to create different amendments, as they do in their jobs.

The task of generating amendments exists of different activities that have to be performed. For more information on how the testers were instructed, see Appendix A

The next tasks were performed for user and acceptance tests in the Netherlands.

Test Tasks

- Create a new document, using templates
- Edit a new document
- Edit the text
- Edit the structure
- Add an article, manually or with copy paste
- Use the print function
- Use the spelling check
- Copy/cut/Paste
- Save a new document
- Rename a documents
- Export a document to PDF
- Create an amending version of an document
- Create an amendment
- Create more then one amendment
- Consolidate amendment.
- Add Comments
- Create References

The technical tests were performed during the same pilot session at the Parliament's Legal Drafting Bureau (Bureau Wetgeving). After installing the editor and during executing the user an acceptance test, the technical part was monitored by experts.

Test Cases

The next amendment were used for testing (see also Appendix A)

- Inserting a revision clause (evaluatiebepaling)
- Inserting a conditonal delegation clause (voorhangprocedure)
- a real Amendment
- fictive Amendment

During performing the different activities the testers were observed en interviewed about their experiences. The outcomes of this observations en interviews are reported. See chapter 5: Findings.



4.2. Italian National Tasks

During the pilot in Italy, there was an opportunity to interview some experts in legal drafting. Experts from the Senate and the Chamber explained that most of the time (80%), document management is already taken care of by either organisations or people. In addition, the Chamber and Senate are in the middle of processes. Because of this their requirements for an editor are:

- a) easy interface, simple and neat;
- b) few buttons, menu, functions;
- c) able to detect in a pre-existing text the inconsistencies in the structure, normative cross-references and respect of the basic rules of the legal drafting;
- d) able to parse the existing parts;
- e) able to insert new parts in a consistency way;
- f) able to repair the existing text;
- g) able to track the versions and to present in a comparative table;
- h) able to consolidate the amendments.

Test Tasks

- Create a new document, using templates
- Merge pre-existing texts
- Automatically parse pre-existing texts
- Repair existing texts with smart tools
- Add new parts in pre-existing texts in a coherent way
- Edit a new document or separate parts thereof: header (number, date, type, authority), preface (motivational part) or structure.
- Manually add a part, title, chapter, subchapter, article, comma, letter or number
- Check internal references
- Check the semantic consistency of the text (article empty, comma without article, etc.)
- Conclusion part (signature, dates, etc.)
- Create references
- Parse an existing document and separate parts thereof: header (number, date, type, authority), preface (motivational part), structure, references or conclusion (signature, dates, etc.).
- Save a new document
- Manage table, images, object embedded in the XML file
- Manage annexes hierarchy
- Export/import document in XML
- Import document in RTF/TXT/WORD
- Export a document into PDF/HTML
- Check the validity respect the MetaLex/CEN and NIR schema
- Semantic check following the Italian Guidelines for Legal Drafting
- Amendments
 - Create an amendment
 - Markup amendments
 - Parser amendments
 - Report of amendments
- Consolidate the modifications
 - Mark-up modifications
 - Create an updated version of a document
 - Create the chain of versioning
 - Create the references with the modifier
 - Annotation of the modifications



- Add comments
- Save frequent references
- Create PDF comparison table that remarks the differences between two versions.

During performing the different activities the testers were observed en interviewed about their experiences. The outcomes of this observations en interviews are reported. See chapter 5: Findings.

Test Cases

Four scenarios were tested:

- mark-up a pre-existing document in XML for sending it from Senate to the Chamber of Deputies (external workflow);
- mark-up a mixed document (partially cut/pasted) coming from the parties to the legal drafting office for a strong revision (internal workflow);
- creation of new document of amendments and application of the amendments approved by the committee to the original document (committee workflow);
- comparison of two texts coming from different steps for printing the synopsis table (internal and external workflow).

4.3. Workflow

The organisations that were involved in the user tests all had their own, existing workflow systems. The workflow that has been established for the SEAL environment is still in its prototype stage, and does not add any major innovations yet. Because of this, it was left out of the final evaluation.

The workflow system was demonstrated during the earlier tests. Furthermore, during interviews and the test period the workflow was created and fine-tuned. In Appendix C the workflow for The Netherlands and Italy is given, together with the XML-files that need to be used in the workflow environment.



5. Findings Dutch Case

This chapter is a list of all observations and feedback gathered during the SEAL pilots at the Bureau Wetgeving (a supporting unit for the Parliament, responsible for legislative drafting).

In general, it can be concluded that most users are enthusiastic about the possibilities the MetaVex editor offers, and it is believed that it can become a very useful tool.

During the test-period, problems and findings were reported, as listed below. One of the major finding was some lacking editing functionality. As can be seen in the list below, most of these problems are solved in the latest version of the Metavex Editor. Templates are also one of the easiest points to create an improvement over existing editors.

Another major obstacle in using the editor is that a lot of input is delivered in non-MetaLex format. An import function is needed before this input can be edited using the editor.

The following chapters describe the different observations, broadly grouped by theme. Behind the observations and results of the test, a possible solution and a status of 'fixing' the problems in a new version of the editor are given. Issues marked as 'Solved' have been eliminated. Those marked as 'In progress' have been partially eliminated, but no full solution has already been implemented. 'Open' issues will not be addressed in the next release of the editor, and will be handled in a later release.

Findings	Possible Solution	Status
Editing		
The layout currently used by the editor does not conform to the official layout.	An option to select the layout should be added	Selection problem solved (we still need the styles)
The editor is lacking an edit-menu and an edit-toolbar.	Add an edit-menu an edit toolbar	Solved
The outline view is lacking spaces.	Create an new outline view	Solved
The template names are not informative enough and should be altered.	Change template names	Solved
The default workspace should be My Documents.	This can be solved during installation probably irrelevant when using the repository of a link can be added to My documents	Solved
There are insufficient templates for the structures. In addition, there are many standard texts that could be included in a template.	Create, in cooperation with the users, new templates including most of the structures and the additional standard text.	In progress
There should also be templates for a cumulative and an alternative list	Acknowledge the presence of the "and/or" in the MetaLex schema, instead of treating it as part of the SentenceFragment.	Solved
The templates should include an insert box where text should be inserted, rather than the dummy	Create a form for filling the templates.	Open



text.		
If there are empty elements in the document that should contain text, this should be made visible.	This could perhaps be done by inserting dummy text/insert boxes whenever an element is left empty.	Solved
It should be possible to paste text from other sources, while retaining structure.	This will probably require a parser and/or a wizard.	Open
When saving a file, an error mentioning an “incorrect name” is given when the extension is forgotten. This is a confusing error message and should be changed.	The problem can be solved by automatically appending the extension.	Solved
It would be nice to rename a document, as the amendments change names during the procedure.	The users name the documents in the file system. We might prefer to separate the “file name” and the “document name”, using RDF. Still, this function may still be preferred when not using the repository.	Open
Pressing space while in an index should have the editor move to the next item.	This is probably difficult to accomplish, as there are examples of indexes that contain spaces. Still, it should be easier for the users to move the cursor out of the index and to the normal text.	Open
Pressing enter when in a SentenceFragmentSubPart or SubPart should end that part, and start a new one (preferably already numbered).	Add functionality to use enter to start a new sentence and part, together with an auto (re) number function.	Solved
Navigation using up and down arrows does not work as expected.	The up and down arrows should move the cursor to the next or previous row, in the same column.	Solved
It is difficult (impossible, perhaps) to change an article from sentences to subparts containing those sentences, or to a list, etc.	Doing this requires knowledge from MetaLex elements and inserting them by hand. Something like the MS Word functionality for list (increase/decrease indent) would really be helpful here.	Open
It should be possible to add comments.	This should probably been done by using annotations from the MetaLex schema	Open
Numbering should be automatic when drafting a new text (but not when making changes to an already-published law).	Add an auto (re) number function	In progress
It would be nice to undo several actions at once (by having a list of all recent edits).	Together with a full operating Undo en Redo functionality, this has to be examined.	Closed
The undo action sometimes works on individual letters instead of words or sentences.	Together with a full operating Undo en Redo functionality, this has to be examined.	Closed
Sometimes, text that should be on the screen disappears.	This bug may be related to the error in the input file.	Solved
It would be nice if the author (the drafter) of a text could be saved somewhere.	This is tricky. It could be stored as metadata, but it should not be present when the document leaves Bureau	In progress



	Wetgeving. Perhaps this is better implemented using the workflow management system.	
--	---	--

Generating Amendments

Wizard		
It would be better if the name of the function could change, something like Generate Amendment or Layout Amendment instead of Export Amendment.	A name change in the code of the editor is recommended.	Solved
The wizard should ask for the Proposer(s) of the amendment, not the Author(s).	Change name of the Author field in the editor	Solved
The question “Type of selecteer bovenliggende map” is not a clear instruction.	This may simply be a problem with the Dutch translation.	Closed
Amending Wizard (sometimes?) shows both Finish and Next buttons.	Only allow ‘finished’ when all necessary data are given,	Closed (the wizard will be replaced soon by a new one)

Layout		
Changes in the title are not displayed correctly (MetaLex elements are used instead of labels).	There is no clear example yet of what such a change should look like.	Open
The generated amendment does not show the date that has been entered in the wizard.	Make a references to the date added.	Solved
The proposers should be named in the amendment as follows: In the title: Single member: His last name, in capitals. Two members: “last name 1 AND last name 2”, in capitals. More members: “last name 1 CS”, in capitals. In the conclusion: all names, in full. Currently, spaces preceding the names are missing in the output.	Change the way of the author is changed. This means that this function has to count the names that are added, so it must be possible to separate the names from the authors.	Solved
The process of generating an amendment is rather tedious, and it would be preferred if we could reduce the number of steps needed. There has also been a request to make it possible to generate several amendments at the same time.	Things I can think of to limit the number of steps: place a button on the toolbar instead of the right-click menu; use RDF to link automatically to the source document; don’t ask for a new file name (but allow the user to save the amendment later). The request for multiple-amendment-generation may be superfluous when the normal procedure is shortened.	Solved
Even when the change to a sentence is very small, the generated amendment will always	Change the part of the changed text in the original amendment, which will return in de amendment.	Open



formulate is as a replacement of the entire sentence. This needs to be more exact.		
--	--	--

Miscellaneous		
<p>Tracking references. If a change is made to a text that is being referred, it would be nice if it could be reported that this might have an impact for the referring law.</p> <p>In addition, a reference “tree” that shows the reference structure would be a nice feature, as it would show for a change in one text, all other text that might be affected. It would also be nice to show the legal basis for each piece of legislation².</p>	These functions ask a full reference structure/tree and by that all references to all documents that are referring to the main document.	In progress (soon)
Automatically updating internal references. It would be nice if all internal references were updated automatically when changing a number of an article or member.	This should be possible as soon as references can be detected. It may lead to some strange behavior when users cut-and-paste a lot, though. External references should of course not be updated, but could perhaps automatically generate the needed amendments for those external texts.	Open
Concept highlighting. It would be nice if a concept in a law could be highlighted, so that if the definition of such a concept changed, it would be easier to check if this has consequences for pieces of text using that concept.	Highlighting should be relatively easy. Some more intelligent actions could be done as soon as the defining sentences can be detected.	Open
It should be possible to import non-MetaLex format, otherwise the tool is useless as long as other organizations still deliver the input in Word format.	Create a parser or copy/paste function to insert other documents.	In progress
Workflow: Certain texts, like an evaluation provision or a temporary regulation, expire after some times. It would be nice to remind people of this just before expiration, as certain follow-up actions may be needed.	This is standard workflow functionality, and is a matter of (advanced) configuration.	Open
After publishing, a document should be read-only, to prevent editing errors.	This should be easily arranged through the authorization functionality.	Open
Authorization is a very important issue, and something the members of Bureau Wetgeving are very worried about. Certain documents should be shared with other organizations, while others shouldn't.	Possible Solution: Create authorized workspaces in the working environment, in which a document can be authorized (read, edit).	Open

² It turns out that a lot of ministerial decrees lack such a legal basis.



If a law has a citation designation, it is not only mentioned in the citation designation provision, but in the title as well (between brackets).	This offers some options for additional functionality, but may also require a slight change to the MetaLex schema.	Open
---	--	------

6. Findings Italy

This chapter is a list of all observations and feedback gathered during the SEAL pilots at the Senate of Republic held on 10th June 2008 and 17th June 2008.

In the meeting of the 10 June the Senate of Republic, the ICT Department and the Legal Drafting Office, were present. The 17 June 2008 the Senate of Republic and Chamber of Deputies and in particular Legal Drafting Office of the Assembly and ICT Departments were present during the tests.

We can distinguish four different types of end-users with different needs.

- Legal Drafting Office of the Assembly: in the 80% of the cases these offices repair existing document coming from parties, deputies, senators, ministries, government, and other institutional bodies. Therefore the main goal is to have a light tool for repairing the existing document and for making up in XML.
- Committee Secretary Office: the main work is to prepare, order and to assemble amendments for the voting session. After a voting session the main task is to apply the amendments to the document. The amendments are not the modifications: the amendments are only textual and no temporal elements affect the application of them; the modification includes also derogation, exception, extension, temporal modifications, etc. The modifications are applied to the official published normative document. The amendments are applied during the legislative lifecycle.
- ICT Department: the ICT Department aims to avoid redundancy of data, to improve the quality of the document, to make each step in the workflow consistent with the others, to provide effective tools for the end-users and saving time, paper, costs and errors.
- General Affair Office of Assembly: the general affair office of assembly need to exchange XML format between the different Houses, institutions, actors. The main task is to produce in output the XML file concerning the internal lifecycle and to import in XML the correspondent outcome of the external bodies. Secondary to make a rapid comparison between the two texts (synopsis table).

So one evident feedback is that not only one tool is needed but four main tools differentiated by type of task:

- repairing and creating - Legal drafting activities
- managing the amendment report and the consolidation of amendments - Committee activities
- comparing different texts coming from different actors, steps, etc.
- parsing the existing document and translate into XML file.

During this test, both the Norma Editor and the xmLegesEditor have been tested. Hence, there are two tables with results, one for each editor. Certain issues have been raised concerning both editors, and are present in both tables. As with the Dutch findings, issues marked as 'Solved' have been eliminated. Those marked as 'In progress' have been partially eliminated, but no full solution has already been



implemented. 'Open' issues will not be addressed in the next release of the editor, and will be handled in a later release.



6.1. Norma Editor

Repairing and creating

Findings	Possible Solution	Status
Editing		
New tools for creating new document, new partitions, new structure with the automatic numbering mechanism	A new tool bar was added with buttons and functionalities for controlling the consistency of the new parts added in the text	Solved
Integration between existing parser tools with creation tools	Two tool bars are used for this purpose with the possibility to insert, delete, renumbering, check the structure of the new partitions	Solved
Possibility to open a form for integrating new parts also from the pre-existing markup module	The form is contextually called in the proper position.	Solved
Numbering should be automatic when drafting a new text	Add an auto (re) number function	Solved
Template for leading the end-user	We have created a template directly with bookmark: only one type for the pilot but the number of template can be increased.	Solved
The template constraints	The template permits to lock the bookmark in order to avoid to delete the relevant part of the template, or with the same metaphor it is possible to un-lock them and the user is free to remove the part of the template useless	Solved
There insufficient templates for the structures. In addition, there are many standard texts that could be included in a template.	Create, in cooperation with the users, new templates including most of the structures and the additional standard text.	In progress
The default workspace should be My Documents.	The editor propose fixed workspace for managing better the operations with the file system	Solved
Interchange of the document with other institutions, actors, users	A shared area is possible on the file system with a mechanism of check-in and check-out during all the phases of the workflow	Solved
Internal reference detecting	It is possible to include an internal reference simply select a part of text and using the right button of the mouse for operating on that.	Solved
Check control of the internal references	A special button was built in order to check the consistency of the internal references after new insertions, deletion, cut-and-paste operations.	Solved
Renumbering internal references		In progress
It should be possible to paste text from other sources, while retaining structure.	This will probably require a parser and/or a wizard.	Open
It is possible to manage annexes	A table of annexes is managed in order to track the hierarchy of the whole package of the normative document.	Solved
It is possible to manage multimedia object	The multimedia object are embedded inside of the XML file	Solved
It should be possible to add comments.	A module for adding annotation makes possible to include in the normative document editing notes or simply	Solved



	comments.	
It would be nice to undo several actions at once (by having a list of all recent edits).	Together with a full operating Undo en Redo functionality, this has to be examined.	Solved
The author, the editor version and the version of the Schema is tracked in the document.	A proper tag is used for archiving in the document the editor version, the author and the Schema version used	Solved
WYSYWYG interface	The editing environment is totally WYSYWYG based and only at the end it is possible to translate in XML	Solved
Export and Import from MetaLex/CEN	Translating and importing from MetaLex/CEN	Solved
Parsing		
Parsing functions on the pre-existing text	Parsing of header, preface, structure, conclusion	Solved
Table markup	Possibility to markup table object	Solved
Parsing of metadata	Some metadata are automatically detected such as date of publication, URI, number, type	Solved
Parsing of references	Cross normative reference are automatically detected and transformed in URI	Solved
Frequent act references	It is possible to save locally the frequently references in order to avoid mistake, errors, and to digit URI	Solved
Thesaurus	It is possible to link a specific thesaurus for the act classification, also EUROVOC it is possible	Solved
Possibility to integrate the markup with proprietary tag	Some legacy system need extra-tag for improving the spider or indexer so it is possible to add proprietary tag with XSLT transformation	In progress

Amendments and Consolidation

Consolidation		
Parsing amendments	Automatic detection of the action of amendment, the proposers, the portions of text, the arguments.	In progress
Markup of amendments	Manually markup of the amendments: action, arguments, dates, notes	Solved
Ordering of the amendments in a report	Possibility on the base of the Regulation of the House to order the amendments for the presentation in the voting session	In progress
Voting amendments module	In the amendments report to annotate the status: rejected, passed, removed, etc.	In progress
Application of the amendments	Module that supports the end-user into the consolidation of the amendments into the original text in semi-automatic way	Solved
Versioning and annotation management	Each step of the lifecycle produces a new versioned file with the date of versioning, the annotations, the amendments applied	Solved

Synopsys and Layout

Comparison table of two versions	It is possible to choice two version Vx	Solved



	and Vz with x<=z and to produce a synopsis with comparison table of the text	
PDF layout	It is possible to produce the PDF layout similar to the official journal of the House	Solved
XHTML	It is possible to produce a transformation in XHTML for any browser through XSLT	Solved
Versioning navigation	It is possible to present all the versions of the same text over time	Solved
Active references and Passive references	It is possible to show the list of the active references (from the document to external documents) and passive references (the link to the document from external documents)	Solved

Server Side

Check the cross reference in the DMS repository	Before the submission it is possible to check the consistency of the cross normative references with the collection of URI present in the DMS. This for avoiding referring repealed norms or incorrect links.	Solved
Check the versioning chain	During the submission to Norma-Server the server validates again the document respect local NIR schema and also checks the consistency with the versioning chain. In case the user submits a wrong version the system block this operation	Solved
Alert for undefined events	In the lifecycle of the legislative acts it is possible to have undefined and unpredictable events, that can affect the legal system. If this happens the server should alert the editor about this	In progress
Modification of modification	In case of modification of modification the system should alert the end user and suggests the proper amendments.	In progress
Support the consolidation of new text	The server support any activities of consolidation	Solved
Simulation environment	It is possible to have a simulation environment for preview checking the legal system during the legal drafting activities. The simulation view can underlines some critical points in the whole legal system. This is not possible to detect with the document view alone.	Solved

6.2. xmLegesEditor

Repairing and creating

Findings	Possible Solution	Status
Editing		
Integration between existing parser tools with creation tools	A function allowing to import plain text or xml standard compliant document portions when drafting a new bill.	Solved
Numbering should be automatic	Add an auto (re) number function	Solved



when drafting a new text		
Template for leading the end-user	A wizard (check list) of the main activities to be accomplished to produce a standard compliant document endowed with all the required information	In progress
The template constraints	A native mechanism to produce standard compliant document with an "a priori" validation.	Solved
There insufficient templates for the structures. In addition, there are many standard texts that could be included in a template.	Create, in cooperation with the users, new templates including most of the structures and the additional standard text.	In progress
The default workspace should be My Documents.	The editor propose fixed workspace for managing better the operations with the file system	Solved
Interchange of the document with other institutions, actors, users	A shared area is possible on the file system with a mechanism of check-in and check-out during all the phases of the workflow	Solved
Internal reference detecting	It is possible to include an internal reference simply select a part of text and using the right button of the mouse for operating on that.	Solved
Check control of the internal references	A special button was built in order to check the consistency of the internal references after new insertions, deletion, cut-and-paste operations.	Solved
Renumbering internal references		Solved
It should be possible to paste text from other sources, while retaining structure.	A validity check of the entering structure is needed.	Open
It is possible to manage annexes	A table of annexes is managed in order to track the hierarchy of the whole package of the normative document.	Solved
It is possible to manage multimedia object	The multimedia object are embedded inside of the XML file	Solved
It should be possible to add comments.	A module for adding annotation makes possible to include in the normative document editing notes or simply comments.	Solved
It would be nice to undo several actions at once (by having a list of all recent edits).	Together with a full operating Undo and Redo functionality, this has to be examined.	Solved
The author, the editor version and the version of the Schema is tracked in the document.	A proper tag is used for archiving in the document the editor version, the author and the Schema version used	Solved
WYSYWYG interface	The editing environment is totally WYSYWYG, while the underlying structure is natively XML.	Solved
Export and Import from MetaLex/CEN	Providing a standard compliant Metalex/CEN template able to cope with the mark-up requirements and workflow phases for a new bill.	Solved
Amending hierarchies as well as plain text portions of a new text, taking track of the amendments	Adaptation of the existing amending functionalities to deal with management of a new bill.	Solved
Parallel view of the drafting bill before and after the amendments insertion.	Using a proper XSL stylesheet to produce such a parallel visualisation of the amending text.	Solved



Possibility to access the knowledge and linguistic resources developed within the DALOS project.	Specific panels presenting the DALOS ontology as well as language-dependent versions of the linguistic resources developed within DALOS	Solved
Parsing		
Parsing functions on the pre-existing text	Parsing of header, preface, structure, conclusion	Solved
Table mark-up	Possibility to mark-up table object	Solved
Parsing of metadata	Some metadata are automatically detected such as date of publication, URI, number, type	Solved
Parsing of references	Cross normative reference are automatically detected and transformed in URI	Solved
Possibility to integrate the markup with proprietary tag	Some legacy system need extra-tag for improving the spider or indexer so it is possible to add proprietary tag with XSLT transformation	In progress

Generating Amendments

Consolidation		
Markup of amendments	Manually markup of the amendments	Solved
Ordering of the amendments in a report	Possibility on the base of the Regulation of the House to order the amendments for the presentation in the voting session	In progress
Voting amendments module	In the amendments report to annotate the status: rejected, passed, removed, etc.	In progress
Application of the amendments	Module that supports the end-user into the consolidation of the amendments into the original text in semi-automatic way	Solved

Synopsys and Layout

Comparison table of two versions	It is possible to choice two version V _x and V _z with x<=z and to produce a synopsis with comparison table of the text	Solved
PDF layout	It is possible to produce the PDF layout similar to the official journal of the House	Solved
XHTML	It is possible to produce a transformation in XHTML for any browser through XSLT	Solved
A module able to show on the browser a specific point in time version of a multi-versioning text.	Using specific stylesheets to show and form to query a specific point in time version of a multi-versioning text.	Solved



7. General Overview

After testing on national level there can be a general overview on the findings during testing and the comments made after testing. On general level, a few remarks can be made, considering the test findings in general and the future international workflow.

After finishing the test phase and gathering the test result, the next table gives an overview of the findings, the number of findings solved and findings that are not solved yet.

7.1. General findings

System	Total	Solved	Open	In progress
MetaVex (Dutch pilot)	38	16	17	5
Norma Editor (Italian pilot)	44	34	1	9
xmLeges Editor (Italian pilot)	34	28	1	5

By using Rapid Application Development a number of the findings are already solved during the test phase. Because of the different test moments, Rapid Application development has even more added value. The potential users/tester, see what is done with their comments, and how development of an editor as tested, works.

It's very useful to take the testers into the process of developments. This creates engagement and, through that, enthusiasm. It makes it easier for the tester to share his ideas and bring up ideas for further development. In the end, this enthusiasm creates added value for all parties concerned.

The use of templates

On both test locations, remarks about the use of templates are made. It is very important that templates exist that are suitable for the situation the editor is used in and contains the right structure and information, needed by the users.

On European level it will be important that there will be templates, suitable for all locations where legislative text drawing is taken care of. This standard implies a European format used for legislative text drawing

Parsing

Parsing is mentioned on both test locations. Parsing is a necessary part of the Norma editor, as it is based on MS Word. For MetaVex and xmLeges, both native XML editors, parsing is not necessary. But it can still be useful for importing documents (from other environments) that have not yet been marked up. It can also assist users by allowing them to enter free text, which can then be parsed to apply the correct mark-up.

The Norma Editor already has elaborate parsing features, to mark-up existing (Italian) Word documents. The Leibniz Center has prototype parsers for detecting structure and references in Dutch legal texts, which are scheduled for incorporation in MetaVex at some later point.



As parsing is very language and jurisdiction specific, they cannot easily be shared for different locations.

Workflow

During the test period the workflow as developed wasn't fully tested. The existence of local workflow systems made testing of the prototype system irrelevant. Connecting to the existing systems was out of scope for this project, and not desired by the users. For both Italy and the Netherlands, there is a good view on the processes that are underlying legislative text drafting.

For European use, there should be a comparison between the different processes, the related documents, and the different statuses a document can get in the process. Another way is to create a European standard. This option is quite hard to accomplish, because of local standards and the historical growth of these standards.

By this, a general workflow model can be created in the Be Informed environment. This creates the possibility.

Usability

Usability of an editor is very important in acceptance by the users. A new editor is adding value only if the (prospective) users will actually be using it to do their daily work. During the tests, the usability items below were marked as important by the end-users:

- Ease of editing. Editing text is one of the biggest tasks performed by legislative text drawers. All necessary functionalities to make editing as easy as possible must be built in.
- Existence of templates
- Creating references, not only in the text that is edited, but also between different documents. References should work two ways. On one side, a reference is a link to another document, on the other side it is a connection between two documents. If something is changed in a document the user should be alerted about the impact of the change on other documents.

Version Management

It's important to construct reliable version management. Amending and consolidating documents can create a complex situation, if version management is not present. The use of a central repository will help keeping track of versions, as users will be required to explicitly check out documents, and to check them in again later. By keeping track of these operations, the different versions can be tracked.

7.2. Future work

One important note can be made after the different tests, held in different countries. It is clear that there are differences between the countries. Each country has its own legislative system, its own legislative processes and its own way of constructing a law.

Splitting a project like SEAL into different editors on national level, contains the risk that the developing and testing is too much on national level, which could mean that the international component does not get the attention that is needed. On the other side, first national research should come up with requirements and experience of users to set up an international European framework, together with a European exchange format in XML. This makes it quite difficult to introduce a general standard for European Legislative text writing. One possible solution is to create an European standard for European laws and

The test phase and test results within the SEAL project, for sure, came up with useful findings and recommendations for further development. The enthusiasm of the



testers about the tool demonstrated that there is a potential gain from this kind of developments and editors.

In general there are some improvements to be made, to make a successful implementation possible. The most important result from the SEAL test phase is that this first group of testers has been made enthusiastic about the possibilities of an editor for legal drafting. Furthermore, they were impressed by the improvements made in developing the different editors. The project has shown that collaborating in the field of legislative is worthwhile. Furthermore, the project stresses that collaboration in the CEN/MetaLex working group pays off. Using CEN/MetaLex collaborative working environments supporting the legislative drafting process at all levels of government has come one step closer.

A complete overview of the results of the tests is given in D4.5 Report on the fluctuations of the values of the success criteria. In the report, an advice for implementation and further activities is included



ATTACHMENTS



Appendix A Test Cases

Dutch Test Case

The text below consist the assignment for the Dutch Test Meeting at February 1st 2008 at the Dutch Parliament at The Haque, for the tasks as mentioned in chapter 4.1. The next two pages give the proposal 30411 that is used during the assignment.

Goedemorgen,

Bijgaand wetsvoorstel 30411 (Regels omtrent instanties die verantwoordelijk zijn voor handhaving van de wetgeving inzake consumentenbescherming (Wet handhaving consumentenbescherming)).

*De bedoeling is dat je voor dit wetsvoorstel een aantal amendementen maakt, zoals:
een evaluatiebepaling
een voorhangprocedure
een van de daadwerkelijk ingediende amendementen die op het stemmingsoverzicht van de EK vindt,
een zelf verzonnen amendement.*

Zou je de problemen die je hierbij hebt op kunnen schrijven, zodat ze het programma kunnen aanpassen aan onze wensen.

BvD

Attachment:
list for voting





Eerste Kamer der Staten-Generaal
Inhoudelijke Ondersteuning

Den Haag, 10 juli 2006

Aan de leden en de plv. leden van de
Vaste Commissie voor Economische Zaken

**OVERZICHT van stemmingen in de Tweede Kamer betreffende
wetsvoorstel:**

30411 Regels omtrent instanties die verantwoordelijk zijn voor handhaving van de wetgeving inzake consumentenbescherming (Wet handhaving consumentenbescherming)

EINDSTEMMING WETSVOORSTEL: het wetsvoorstel is op 7 juli 2006 aangenomen door de Tweede Kamer. SP, GroenLinks, PvdA, D66, CDA, ChristenUnie en SGP stemden voor.

AANGENOMEN EN OVERGENOMEN AMENDEMENTEN

Artikel 2.6

14 (Algra)

Het is van de belang dat de Consumentenautoriteit niet inhoudelijk verwickeld raakt in een onderhandelingsproces tussen bedrijven en gedupeerde consumenten. Immers de onafhankelijkheid van deze toezichthouder mag op geen enkele wijze in gevaar komen. Daarom is het wenselijk dat de Consumentenautoriteit samen met betrokken partijen een voor allen aanvaarbare derde aanwijst die de onderhandelingen kan begeleiden die leiden tot het opstellen van een overeenkomst die de afwikkeling van de massaschade regelt. Met dit amendement wordt bereikt dat de Minister ervoor zorgt dat de Consumentenautoriteit daadwerkelijk een onafhankelijk derde inschakelt. Teneinde alle consumenten die gedupeerd zijn aanspraak te kunnen laten maken op die schadevergoeding, blijft de Consumentenautoriteit op grond van de wet bevoegd om de genoemde overeenkomst door de rechter algemeen verbindend te laten verklaren.

Aangenomen. Voor: PvdA, Groep Wilders, VVD, CDA, ChristenUnie, SGP, LPF, Groep Nawijn en Groep Van Oudenallen



VERWORPEN, INGETROKKEN EN/OF VERVALLEN AMENDEMENTEN

Artikel 2.7, eerste lid

Artikel 3.2b

Artikel 9.3, onderdeel e

10→13 (Crone)

Dit amendement betreft een technische aanpassing van het eerder ingediende amendement op stuk nummer 10. Er is in het wetsvoorstel gekozen voor overwegend privaatrechtelijke handhaving, waarbij de Consumentenautoriteit (hierna: CA) zelf procespartij wordt en gaat procederen tegen de overtreder. Voor slechts twee door de CA te handhaven richtlijnen en corresponderende wetten (Colportagewet en Prijzenwet) is gekozen voor de (veel effectievere) bestuursrechtelijke handhaving, zoals we die kennen van de NMa en de AFM. Privaatrechtelijke handhaving blijkt echter niet altijd even effectief te zijn, en kan juridisering en eindeloze processen in de hand werken. Daarom moet de bestuursrechtelijke handhaving ook van toepassing zijn op de overige wetten waarop wordt gehandhaafd. Dit betekent dat de consumententoezichthouder bestuurlijke boetes moet kunnen opleggen en met dwangsommen de handhaving van het consumentenrecht moet kunnen afdwingen. Op deze manier wordt het doel van de wet, namelijk een effectieve consumentenbescherming, bevorderd. Uiteraard staan besluiten van de CA open voor de gebruikelijke mogelijkheden voor beroep en bezwaar. De weg van de verzoekschriftenprocedure blijft voor de CA ook openstaan als alternatief voor de bestuursrechtelijke handhaving. Ook voor consumentenorganisaties blijft de verzoekschriftenprocedure openstaan, zoals die in het wetsvoorstel al voorzien was.

Ingetrokken

Artikel 2.6

7 (Örgü)

De Consumentenautoriteit heeft in het wetsvoorstel de bevoegdheid gekregen de rechter te verzoeken om een overeenkomst ter afwikkeling van een massaschade verbindend te verklaren voor personen aan wie de schade is veroorzaakt. Het amendement strekt ertoe deze bevoegdheid te schrappen.

De Consumentenautoriteit dient niet de rol over te nemen van private partijen die de belangen van de consument behartigen – zoals de consumentenbond – en die daartoe onder andere over de mogelijkheden van de Wet collectieve afwikkeling massaschade en de Wet collectief actierecht beschikken.

Verordening 2006/2004 noopt niet tot de maatregel de Consumentenautoriteit een positie te geven in het kader van de Wet collectieve afwikkeling massaschade. Dit amendement brengt het wetsvoorstel daarmee in lijn met het recente kabinetsbesluit om nationale aanvullingen bovenop Europese regels zoveel mogelijk te schrappen.

Verworpen. Voor: Groep Wilders, VVD, LPF, Groep Nawijn en Groep Van Oudenallen

Artikel 7.2

8 (Örgü)

Dit amendement strekt ertoe consumentenorganisaties niet per definitie aan te wijzen als belanghebbende bij besluiten krachtens de Wet handhaving consumentenbescherming.



De AWB geeft de rechter reeds de ruimte om afhankelijk van de omstandigheden van het geval te beoordelen of consumentenorganisaties belanghebbende zijn bij besluiten krachtens de Wet handhaving consumentenbescherming.

Een belanghebbende in de zin van de Awb is namelijk degene wiens belang rechtstreeks bij een besluit is betrokken (art.1:2, lid 1, Awb). Ten aanzien van rechtspersonen worden als hun belangen mede beschouwd de algemene en collectieve belangen die zij krachtens hun doelstellingen en blijkens hun feitelijke werkzaamheden in het bijzonder behartigen (art. 1:2, lid 3, Awb). De indiener is derhalve van mening dat artikel 7:2 van het wetsvoorstel overbodig is.

Bovendien vereist Verordening 2006/2004 niet dat consumentenorganisaties worden aangewezen als belanghebbende bij besluiten krachtens de Wet handhaving consumentenbescherming. Dit amendement brengt het wetsvoorstel in lijn met het recente kabinetbesluit om nationale aanvullingen bovenop Europese regels zoveel mogelijk te schrappen.

Verworpen. Voor: Groep Wilders, VVD, LPF, Groep Nawijn en Groep Van Oudenallen

MOTIES

15 (Örgü) over leidende principes in de handhavingspraktijk van de Consumentenautoriteit

Afgevoerd van de agenda

11 (Algra) om te komen tot een ConsumentenAutoriteit met een slanke organisatie en een formatie van maximaal 10 fte's

Ingetrokken

16 (Örgü) over bundeling van de voorlichting aan consumenten door de verschillende departementen

Verworpen. Voor: VVD en Groep Wilders

19 (Örgü) over onderbrenging van het consumententoezicht bij een bestaande toezichthouder

Verworpen. Voor: VVD, Groep Wilders, LPF en Groep Nawijn



Appendix B Evaluation Questionnaire

After testing the users were asked to fill in the evaluation Questionnaire which contained next question. Each question can be rated as: *Good, Better, Same, Worse* or *Poor* by which the editor is compared to the existing situation.

Society

1. Efficiency of legislative text?
2. Effectiveness of legislative text?
3. Insight in the negative side effects of legislative text?
4. Consistency in legislative text?
5. Clarity in legislative text?
6. Acceptability of legislative text?
7. Legitimacy of legislative text?
8. Feasibility and manageability of legislative text?
9. Simplicity of legislative text?
10. Social and political feasibility of legislative text?
11. How do you consider the level of accessibility of legislative text using the new application?
12. How do you consider the level of transparency in the legislation processes, using the new application?

Organizational

1. How do you consider the level of operability between applications using the new application?
2. How do you consider the level of operability between different organizations and body using the new application?
3. How do you consider the level of control of the workflow using the new application?
4. How do you consider the level of compliance to organizational standards using the new application?
5. How do you consider the level of efficiency and effectiveness using the new application?
6. How do you consider the level of costs using the new applications

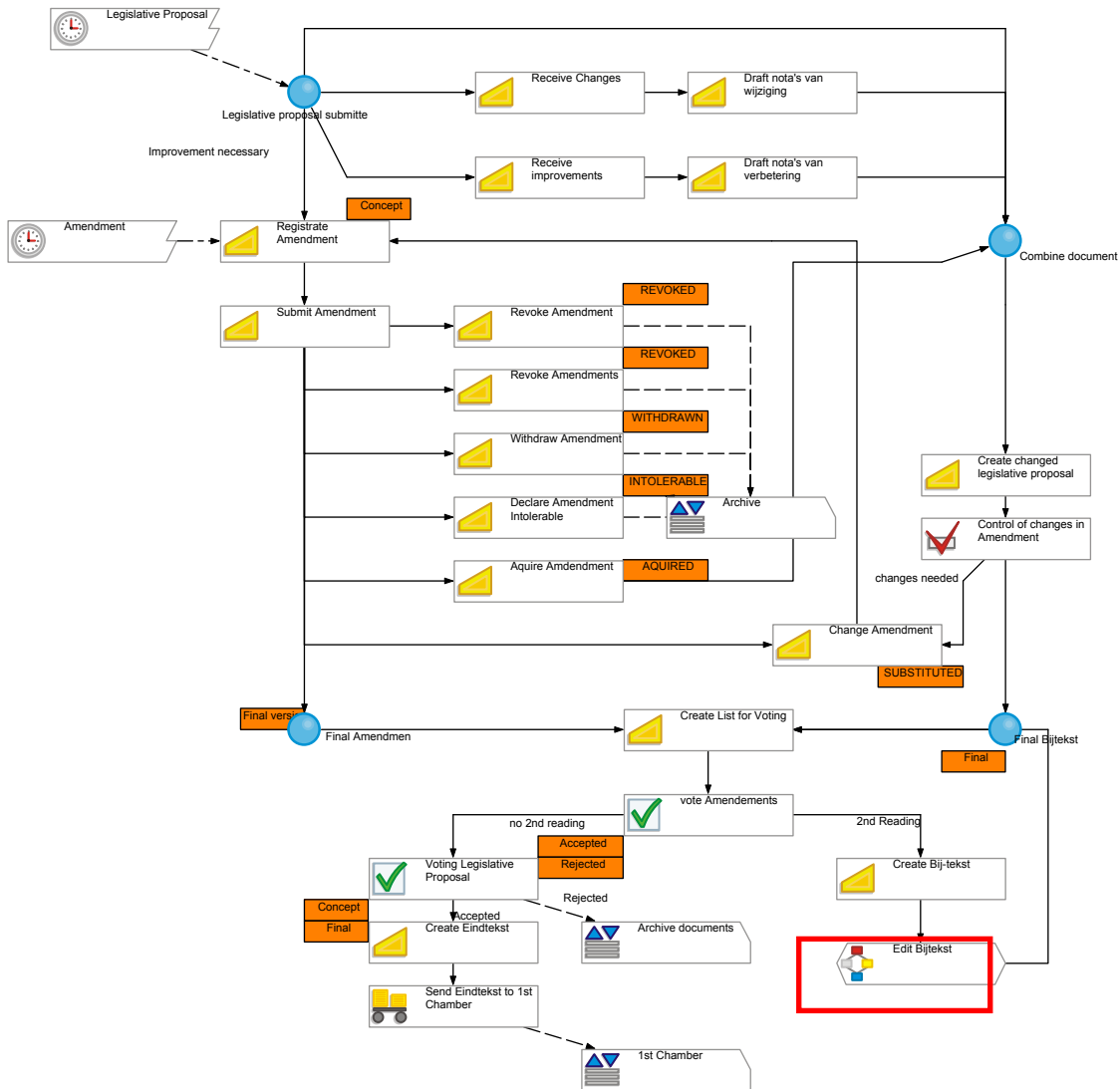
User

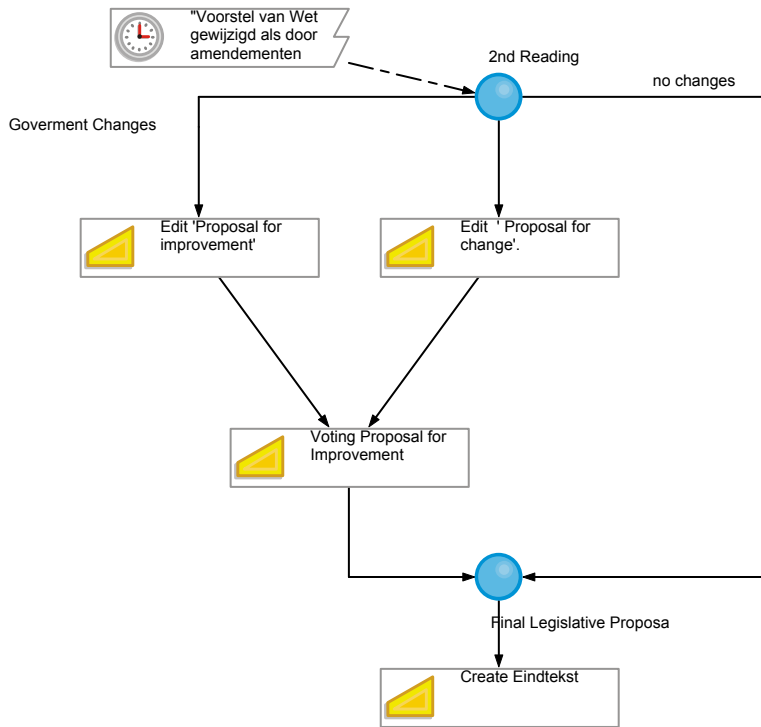
1. How do you consider the level of writing speed using the new application?
2. How do you consider the level of interaction with other users using the new application?
3. How do you consider the level of interaction speed with other users using the new application?
4. How do you consider the level of accuracy using the new application?
5. How difficult is it using the new application, in your opinion, to comply with organizational standards?
6. How difficult is it using the new application in your opinion, to organize legislative text?
7. How do you consider the level of quality in legislative texts using the new application?
8. How difficult is it using the new application, in your opinion, to comply with organizational standards?



Appendix C Workflow

Workflow Dutch case





XML Dutch case

```
<types>
  <type>Amendment</type>
  <type>Bill</type>
  <type>Bill Improvements</type>
  <type>Bill Changes</type>
  <type>List of Voting</type>
  <type>Changed Bill</type>
  <type>Final Version</type>
</types>

<priorities>
  <priority>Low</priority>
  <priority>Medium</priority>
  <priority>High</priority>
</priorities>

<statustransitions>
  <statustransition from="">
    <allowed>CONCEPT</allowed>
    <allowed>SUBMITTED</allowed>
  </statustransition>
  <statustransition from="CONCEPT">
    <allowed>SUBMITTED</allowed>
  </statustransition>
  <statustransition from="SUBMITTED">
    <allowed>REVOKED</allowed>
    <allowed>WITHDRAWN</allowed>
    <allowed>INTOLERABLE</allowed>
    <allowed>ACQUIRED</allowed>
    <allowed>SUBSTITUTED</allowed>
    <allowed>ACCEPTED</allowed>
  </statustransition>
  <statustransition from="ACCEPTED">
    <allowed>REJECTED</allowed>
    <allowed>ACCEPTED</allowed>
    <allowed>FINAL</allowed>
  </statustransition>
</statustransitions>
```



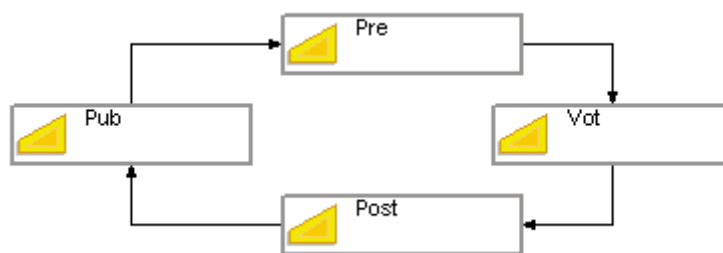
Workflow Italian case

Inside of the Italian legislative process we can find two main macro-processes:

- **Internal process:** each house produces several steps: each house has several internal activities, not all are machine computable so the workflow should be hybrid: some steps done with tools and computer, some steps by hand on paper. The tools should support a hybrid organization and process permitting to pass from paper to digital format and vice versa.
- **External process:** exchanging document between institutions. At the end of each internal process the institutions (Senate and Chamber, sometime also the Government) fix the version and sent it to the other House. This moment is official and it is published in each House Official Journal as a sort of check-in, check-out process. The publishing is done in a comparative format: one column shows the previous outcome and the second column shows the new income with the differences underlined.

Each macro-process is divided in 4 macro-steps:

- **Pre-voting activities:** any activities done before the voting of the body deputed to approve the text. In this phase the legal drafting office works strongly for readjusting and for repairing the text in order to improve the quality, to add the amendments, to produce new text. During the communication with the other House the legal drafting office must deduct the modification done in the other House and to mark-up them.
- **Voting activities:** any activities performed inside of the body deputed to approve the text. In this phase new amendments can arise from the government, from the president of the House and from the president of the committee.
- **Post-voting activities:** after the voting the legal drafting must apply all the new amendments and the old one approved on the text.
- **Publishing:** the outcome of the post-voting is called "messaggio" and it is published officially in the House Official Journal. This document becomes the input for the next cycle in the other House.



Following to these characteristics we focus our attention on three main scenarios:

- a. mark-up a pre-existing document in XML for sending it from Senate to the Chamber of Deputies (external workflow);
- b. mark-up a mixed document (partially cut/pasted) coming from the parties to the legal drafting office for a strong revision (internal workflow);
- c. creation of new document of amendments and application of the amendments approved by the committee to the original document (committee workflow);



- d. comparison of two texts coming from different steps for printing the synopsis table(internal and external workflow).

