DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Formalizing and automating regulatory document versioning

Thien-An Huynh

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Formalizing and automating regulatory document versioning

Formalisierung und Automatisierung der Versionierung gesetzlicher Dokumente

Author:	Thien-An Huynh
Supervisor:	Prof. Florian Matthes
Advisor:	Mahdi Dhaini
Submission Date:	15.04.2023

I confirm that this master's thesis in information systems is my own work and I have documented all sources and material used.

Munich, 15.04.2023 Thien-An Huynh

Acknowledgments

First and foremost I would like to thank Professor Florian Matthes for providing me with the opportunity to write my thesis at his chair. In addition to this, I greatly appreciate him taking a personal interest in the subject matter, and for taking the time to discuss the relevant outcomes for this thesis.

Further, I would like to especially thank my advisor at the SEBIS chair, Mahdi Dhaini, for his patience, enthusiasm, and commitment to me and the thesis result. I do not take for granted the weekly meetings, taking up a lot of time and effort, which have helped me greatly with direction and purpose.

In addition to the SEBIS chair and TUM, I would also like to thank the people at Certivity:
Bob for his deep expertise in the regulatory field, answering questions with great patience, and for providing most of the data without which this thesis would not have been possible.
Sergii for being a constant source of encouragement and reason.
Nico for always being supportive, understanding, and putting my personal success first.

Abstract

Legal consolidation is the act of applying modificatory provisions to a target document in the correct order and manner. In current practice, this process is very time-consuming and error-prone, in large part due to the fact that, at the moment, this process is performed exclusively by hand.

In the current state of research, there have been numerous attempts to automate this process by way of leveraging natural language processing techniques and machine learning models in order to automatically parse these modificatory provisions (often only provided in unstructured natural language formats) into an automatically executable formalized format. However, researchers have not yet been able to come up with a reliable method of semantic information extraction or annotation for this particular use case and the problem of automatically parsing these instructions remains largely unsolved. Most of these research projects also purely focus on the information extraction part of the consolidation process, only assessing the correctness of the extracted information and choosing to forgo the automatic application of these instructions to the targetted document. As a final note on the current research, there currently exists no research on regulatory documents in the English language in the context of automatic legal consolidation. In fact, the vast majority of research papers in this field are conducted by Italian researchers.

In this thesis, the goal is to research and develop requirements to formalize modificatory provisions. This format will be based on existing research but specifically focused on amending documents released by the UNECE as well as the federal government of the US. This format is designed to be human-readable and -writable as developing a system that is able to automatically parse natural language text into this format would increase the scope of this thesis to an unreasonable level. In addition to this, a consolidation engine reference implementation is developed, that is able to apply a subset of these modificatory provisions automatically to a target document.

Lastly, both the formalization format and the consolidation engine were evaluated by manually converting unstructured natural language modificatory provisions into this machine-executable format and feeding it to the consolidation engine reference implementation. The resulting outcome was compared to consolidated documents, consolidated by regulatory experts at Certivity, serving as the ground truth. The machine-executable format was evaluated in terms of its expressiveness and whether it is able to accurately model all modificatory provisions from the UN and US data sets. The consolidation engine reference implementation was evaluated on the correctness of the resulting consolidated document, provided that the

modificatory provision has been accurately converted. Lastly, interesting edge cases and modificatory provisions are documented and categorized, which could cause issues for a fully automated consolidation system.

Overall the findings in this thesis indicate that basic replacements, insertions, and deletions of regular chapters are quite trivial to automate. However, the amending documents from the data set contain a lot of implicit information (possible to interpret as a human, but hard for a machine), unusual edge cases, and mistakes made by the regulatory body, preventing the full automation of the consolidation process even with 100% correct conversion into a machine-executable format. Nevertheless, there are lots of promising areas for future research in order to further increase the reliability of a fully automated consolidation system and interim results are quite promising both from existing research and this thesis.

Contents

1. Introduction

1.1. Motivation

As the pace of product development in our progressively more digitized and globalized world keeps accelerating, it is only natural that regulatory supervision has to expand and adapt at a similar pace and in accordance with these new developments. As such, it is becoming an increasingly difficult task to keep up and comply with changes made in regulatory documents, especially for smaller companies with lesser resources.

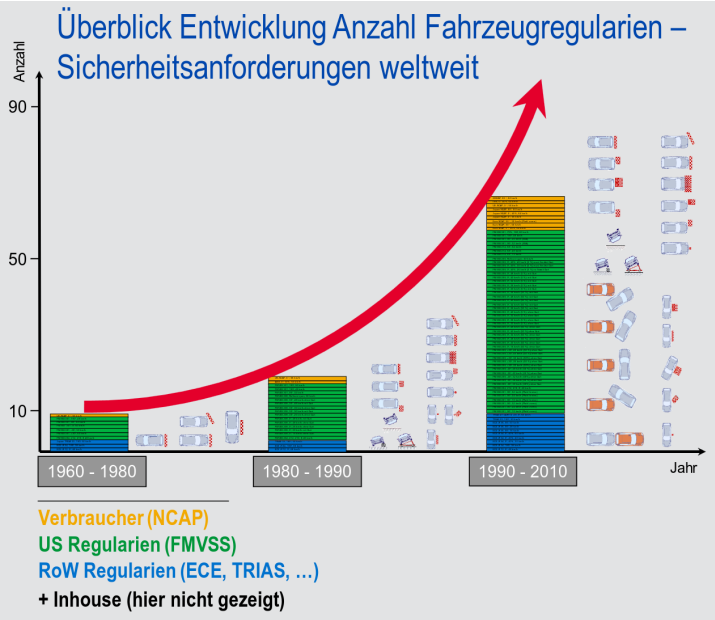


Figure 1.1.: A visualization of the increasing number of regulations over time, provided by TÜV Süd. The translated title reads: "Overview of the number of vehicle regulations over time - safety requirements worldwide".
Yellow = New Car Assessment Program
Green = Federal Motor Vehicle Safety Standards (US)
Blue = Rest of World: United Nations Economic Commission for Europe, Traffic Information Accessibility Standard, ...

Further exasperating the difficulty of staying compliant with the newest regulation versions

is the fact that updates to regulatory documents are usually only released in an imperative manner:

Instead of releasing a complete, updated version of the new document version, regulatory bodies will instead release an amending document [2.1.2], containing modificatory provisions [2.1.4]. These modificatory provisions are instructions that explain in detail what changes need to be performed on the target document in order to create the newer, updated version. In a process called *legal consolidation*, these modificatory provisions can then be applied to the original document version. In practice, this process is exclusively performed by hand and tends to be time-consuming and error-prone in large part due to its non-automated nature.

Automation of legal consolidation is therefore important for several reasons:

- **Speed:** Automation can significantly speed up the legal consolidation process, allowing organizations to be able to focus more on the actual development of their product and to have clear, up-to-date requirements that can be traced back to the specific regulatory provision.
- **Cost-effectiveness:** Automation can also reduce the costs associated with legal consolidation. Automated systems can process large amounts of information more quickly and at a significantly lower cost in comparison to manual labor.
- **Consistency:** Automated systems can ensure consistency in the consolidation process, which can be especially important when dealing with large amounts of data and complex legal documents. Automated systems can be programmed to follow a specific set of rules and guidelines, ensuring that the final consolidated document is accurate and consistent.
- **Auditability:** Automated systems can also provide a clear and auditable record of the consolidation process, which can be important for compliance and regulatory purposes. With manual legal consolidation, a lot of decisions are made implicitly, based on "common sense" and are usually not documented, therefore making them hard to retrace and understand in case of a faulty consolidation result.

1.2. Objectives and Research Questions

Since there already exist various papers on the automatic annotation of modificatory metadata (usually performed by larger research teams and in collaboration with legal researchers and experts), the focus of this thesis is placed on the automatic application of manually created representations of modificatory provisions. In order to gain a deeper understanding of the requirements, complications, and methodologies of automatic change application during legal consolidation, requirements for a machine-executable formalized format are elicited. This format should be able to model all modificatory provisions in our data set, containing regulatory documents released by the United Nations Economic Commission for Europe

(UNECE) as well as the federal government of the US. This format is designed to be human-readable and -writable, so as to exclude automated information extraction from unstructured natural language data from the scope of this thesis.

Additionally, a consolidation engine reference implementation is developed, which is able to take base documents [2.1.1] and automatically apply modificatory provisions to them. Lastly, the machine-executable format is evaluated for expressiveness, and the consolidation engine reference implementation for correctness.

Overall, the set of desired artifacts resulting from this thesis is the following:

- Set of necessary consolidation engine operations to model all modificatory provisions in the data set
- Set of required metadata fields for the representation of base documents & modificatory provisions in order to perform automatic consolidation
- Reference implementation of a consolidation engine, that supports as many of the aforementioned operations as possible
- Classification of problematic consolidation engine operations/modificatory provisions as areas for future research

As such, the research questions are formulated accordingly:

RQ1: What is the minimum set of consolidation engine operations needed to model all modificatory provisions in the data set?
An interface for a consolidation engine should be developed, that exposes functions that are necessary to model all modificatory provisions in the data set.

RQ2: What is the minimum set of metadata fields needed in the representation formats of base documents and modificatory provisions in order to perform automatic consolidation?
For the consolidation engine to perform automatic consolidation, certain metadata fields will need to be extracted from the text and marked up in the corresponding representation formats for easy machine readability and accessibility. The minimum set of this metadata is to be determined as part of this thesis.

RQ3: How accurate are the automatic change applications performed by the consolidation engine reference implementation?
In order to evaluate the consolidation engine's ability to apply modificatory provisions automatically, an experiment is constructed, in which the automatically consolidated documents are compared against consolidated documents which were consolidated manually by regulatory experts.

2. Fundamentals

2.1. Important Definitions

For the purposes of this thesis and in the context of the regulatory domain, the following chapters define some important terms, that will be used further in the thesis.

2.1.1. Base Document Version

Any document version will be referred to as a base document version, that contains all up-to-date provisions of its current version and has been approved by the corresponding regulatory body. Any changes made to it must also be approved by the regulatory body before they can take effect. Subsequent versions of the document that contain changes or revisions to this base document version are sometimes referred to as "amended" or "revised" versions.

2.1.2. Amending Document

In contrast to documents depicting a base document version, amending documents do not contain every provision of its current version. Rather, they contain the necessary modificatory provisions that are to be applied to a target version in order to create a newer, updated version of the regulation.

2.1.3. Target Document

Amending documents will state which document version they want to target with changes. During this thesis, this document/document version will be referred to as the target document of this amending document.

2.1.4. Modificatory Provision

Modificatory provisions refer to clauses within a document that are declaring changes to the content of another document version. These provisions are typically used to update and clarify the original content of the document or to add new information or requirements.

In the context of this thesis, one modificatory provision can contain multiple different *amending instructions* as further illustrated in the examples below.

```
Insert a new paragraph 1.5., to read:

"1.5.      The definitions given in Regulation No. 48 and its series of
           amendments in force at the time of application for type approval
           shall apply to this Regulation."

Paragraph 2.2.2., amend to read:

"2.2.2.    a brief technical description including, in the case where headlamps
           are used to produce bend lighting, the extreme positions according
           to paragraph 6.2.9. below;"
```

Figure 2.1.: Example of two modificatory provisions issued by the UNECE. They each contain exactly one amending instruction.

■ 4. Amend § 571.203 by revising paragraph S2 and removing and reserving S3.

The revision reads as follows:

§ 571.203 Standard No. 203; Impact protection for the driver from the steering control system.

* * * * *

S2. *Application.* This standard applies to passenger cars and to multipurpose passenger vehicles, trucks and buses with a gross vehicle weight rating of 4,536 kg or less. However, it does not apply to vehicles that conform to the frontal barrier crash requirements (S5.1) of Standard No. 208 (49 CFR 571.208) by means of other than seat belt assemblies. It also does not apply to walk-in vans or vehicles without a steering control.

Figure 2.2.: Example of one modificatory provision issued by the federal government of the US. It contains one DELETE operation and one AMEND operation.

2.1.5. Legal Consolidation

Legal consolidation is the process of applying modificatory provisions to a base document version in the correct order, thereby combining multiple legal entities into one cohesive document.

2.1.6. Consolidated Document

The resulting artifact of legal consolidation will be referred to as a consolidated document. Consolidation is rarely performed by the governing body. Rather, regulatory document

users are often reliant on external parties providing this service. Even if the governing body does release a consolidated document for a particular version, it is usually presented to be non-binding and, in case of conflicting information, the information from the original modificatory provision is the valid one.

2.1.7. Consolidatory Metadata

During this thesis, textual content provided by a document will be referred to as *data*. In contrast to this, metadata is information that is not explicitly stated in the original text but which can be inferred from it either by humans or machine learning and rule-based algorithms. Examples of metadata for any particular modificatory provision might include:

- **Modificatory Provision Borders**: demarking where each modificatory provision starts and ends.
- **Target Chapter**: represents the target chapter of a modificatory provision.
- **New Content**: for provisions that include REPLACE or INSERT instructions.

2.1.8. Structured vs. Unstructured Data

In the context of this thesis, unstructured data are legal documents, which are saved in formats that do not inherently support machine readability on a semantic level. Usually, this refers to PDF documents, although some DOCX documents can also be considered unstructured since the use of formatting templates (e.g. Heading1, Heading2, Paragraph, etc.) is purely optional and would be needed in order to correctly model the hierarchical structure of the document.

In contrast, structured data is defined as regulatory document representation formats, which clearly and correctly model the ontological nature of chapter relationships. Additionally, most structured document representation formats offer the ability to mark up certain parts of the text with extensible metadata for machine-readability on a semantic and structural level.

2.1.9. Document Node

Because of the inherent hierarchical tree structure that document chapters have, document representation formats will usually be based on a tree structure as well. As such, the smallest possible element in such a tree structure will be referred to as a document node.

2.1.10. Non-chapter Document Entity

In the context of a legal document, chapter entities are typically major sections of the document that are given their own chapter numbering and are often used to group related

sections or provisions together. Chapter entities are often used to help organize and navigate the document, and may be referred to by their chapter number or name.

Non-chapter entities, on the other hand, are typically smaller components of the document that are not given their own chapter numbering. These entities may include footnotes, tables, images, graphs, definitions, or introductory paragraphs. They are typically referenced via the chapter to which they belong and usually have their own numbering scheme. As an example of localized numbering, footnote numbering is usually reset in-between chapters so there might exist footnotes 1-3 for chapter 1.2. which are distinct from footnotes 1-3 in chapter 2.5.

2.2. Regulatory Document Versioning

In general, regulatory documents are updated not by releasing a complete document in its newest version but rather by issuing modificatory provisions: instructions on how to modify its previous version to create the current, newest version. This chapter further illustrates in detail, the amendment processes of the United Nations Economic Commission for Europe (UNECE) as well as the federal government of the USA.

2.2.1. Amendment Process of the UNECE

The amendment process for UNECE follows certain conventions and agreements and usually involves the following steps:

1. **Proposal:** A member state or group of member states can propose an amendment to a UNECE convention or agreement. These are generally released as delta documents [2.1.2].
2. **Review & Adoption:** The proposed amendment is reviewed by the relevant UNECE working group or committee to assess its consistency with the existing convention or agreement and its potential impact on other member states. If the proposed amendment is accepted by the working group or committee, it is sent to the UNECE commission for adoption.
3. **Adoption:** The commission can adopt the amendment by a two-thirds majority of the member states present and voting.
4. **Ratification:** After the amendment is adopted by the commission, it must be ratified by the member states in order to enter into force. The process of ratification varies depending on the specific convention or agreement, but it usually involves the depositing of an instrument of ratification with the UNECE secretariat.
5. **Entry into force:** The amendment enters into force on a date specified in the convention or agreement, usually 30 days after the deposit of the required number of instruments of ratification.

It's worth noting that the process can vary depending on the specific convention or agreement and the nature of the proposed amendment, but in general, it follows the aforementioned steps.

2.2.2. Amendment Process of the Federal Government of the US

The amendment process for regulatory provisions of the federal government of the United States largely follows the same structure as that of the UNECE and, in general, includes the following steps:

1. **Proposal:** An agency of the federal government can propose an amendment to a regulatory provision, usually through a notice of proposed rulemaking (NPRM) published in the Federal Register. The NPRM typically includes a description of the proposed amendment and an explanation of its rationale.
2. **Comment period:** The public is given an opportunity to comment on the proposed amendment during a specified comment period, usually 30 to 60 days.
3. **Review:** The agency reviews and considers the comments received from the public, and may make changes to the proposed amendment as a result.
4. **Final rule:** After the review process, the agency publishes a final rule in the Federal Register, which includes the final text of the amendment and a summary of the comments received and the agency's response.
5. **Effective date:** The amendment becomes effective on a date specified in the final rule, which is usually 30 to 60 days after the publication in the Federal Register.

It is worth noting that the process can vary depending on the specific regulation and the nature of the proposed amendment. Some proposed changes may require a longer process, such as a cost-benefit analysis, environmental impact statement, or review by other agencies.

Additionally, the process can be subject to judicial review, where citizens or organizations can challenge the legality of the new regulatory provision in court. Also, Congress can use the Congressional Review Act to review and potentially disapprove recent regulations issued by the executive branch.

2.3. Legal Document Representation Formats

From a machine-readability standpoint, legal documents are often originally released in a very unstructured format, usually PDF. In order to leverage the advantages of digital tooling and machine learning, these documents have to be converted into a representation format that is better able to model all aspects of regulations.

The XML format suits itself well as a basis for such a standard, as it is able to easily represent the hierarchical tree structure of chapters inherent in legal texts. Importantly, in addition

to this, the XML format is highly extensible, allowing for the addition and constraint of metadata fields and custom elements.

While it is difficult to provide exact values for the adoption rate of different technical standards for the representation of legislative documents, *Akoma Ntoso* and *Norme in Rete* were the only standards found in the literature that discusses the automation of the legal consolidation process and seem to be the most popular ones (although NIR is exclusively used in Italy). HTML is also used by a number of regulating bodies, including the EU to release their regulatory documents. These formats are thus further discussed in the following.

2.3.1. NIR

Norme in Rete (NIR) is a standard established for the representation of Italian legislation, first proposed in 2001 by the Italian National Center for Information Technology in the Public Administration in conjunction with the Italian Ministry of Justice.

Being XML-based, the standard defines a hierarchical tree structure for legislative documents, including elements such as the document itself, acts, sections, articles, and paragraphs. Similarly to HTML, it also specifies a set of metadata elements for describing the document, such as the title, date, and language.

2.3.2. Akoma Ntoso

Akoma Ntoso is a technical standard for the representation of legislative documents, developed by the United Nations *Department of Economic and Social Affairs* (UN DESA) and the *Organization for the Advancement of Structured Information Standards* (OASIS). The goal of the standard is to provide a common format for the representation of legislative documents, in order to facilitate their exchange and reuse across different regions and platforms.

Akoma Ntoso is based on XML (eXtensible Markup Language) and is designed to be flexible and extensible, and can be used for a wide range of documents, including statutes, bills, regulations, and parliamentary proceedings.

2.3.3. HTML

Hypertext Markup Language (HTML) is very similar to XML in syntax and structure. Both of them define a hierarchical tree structure as the underlying data structure with tree elements containing a set of extensible metadata attributes. In contrast to XML however, which is largely use-case agnostic (therefore enabling the creation of domain-specific dialects), HTML was specifically designed to be displayed by web browsers. Although technically (and similarly to XML), the author of an HTML document does not specify the rendering style of the resulting document, there are well-established conventions and expectations bound to specific tags and the way they are displayed in the end. Some examples are:

- <b/> = **bold**
- <u/> = underline
- <s/> = ~~striketrough~~
- <sup/> = super^{script}

3. Related Works

3.1. Systematic Literature Review

In this chapter, we discuss related works concerning either the general formalization of regulatory documents into machine-readable formats, annotation of legal documents with relevant metadata, and automation of the consolidation process.

In order to determine the current state of research as well as to determine any research gap, relevant literature was identified using a systematic literature review. During the initial stage of this review the search queries were determined in the following way:

Brainstorm general keywords
consolidation, amendment, modification, change, version control, versioning, legal, law, norms, regulations, legislations, formalization, automation, NLP, natural language processing, machine learning

Stem words to improve generalizability
consolidat*, amend*, modif*, change, version control, version*, legal, law, norms, regulat*, legisl*, formaliz*, automat*, NLP*, natural language processing, machine learning

Categorize based on semantic meaning
Describing consolidation and the modificatory nature of it:
- consolidat*, amend*, modif*, change, version control, version*

Describing the object of interest, i.e. regulations:
- legal, law, norms, regulat*, legisl*

Describing the use of ICT and computer science to enhance the consolidation process:
- formaliz*, automat*, NLP*, natural language processing, machine learning

Build final search string with AND, OR clauses
Query 1: ("consolidat*" OR "amend*" OR "modific*" OR "change" OR "version control" OR "version*")
AND
Query 2: ("legal" OR "law" OR "norm" OR "regulat*" OR "legisl*")
AND
Query 3: 5mm("formaliz*" OR "automati*" OR "NLP*" OR "natural language processing" OR

"machine learning")

Every permutation of the resulting query set was used in order to perform searches within the literature search engines. For this literature review in particular, the following search engines were used:

- ACM Digital Library
- Web of Science
- Scopus
- Google Scholar

Further, exclusion criteria were defined in order to determine the relevance of each paper to our research topic. They read as follows:

- Publication date: Released before the year 2000
- Domain: paper has no relation to both computer science and the legal field
- Object of interest: Objects of interest are not legal documents, more specifically norms and regulations (contracts, court decisions, etc. were excluded)
- Consolidation/versioning topic relevancy: Title and abstract make no mention of consolidation, amendments, modifications, versioning, etc.

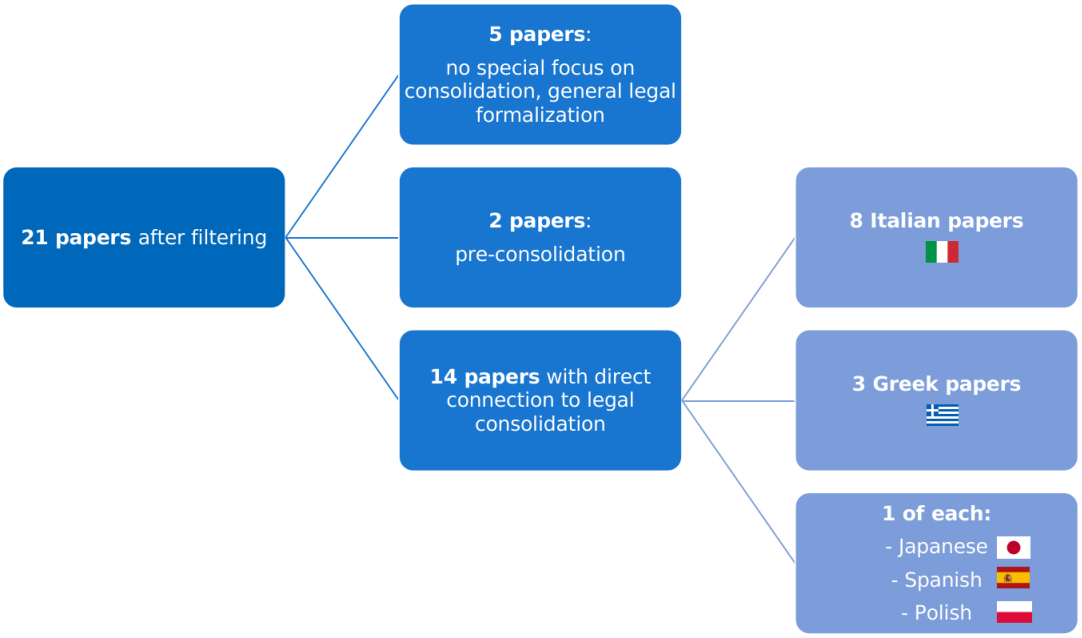
Even after tweaking the search string multiple times, the false positive rates (i.e. found with the search string but meeting exclusion criteria) from search engines regarding relevancy were quite high with a relevancy ratio of less than three percent, among the first 50-100 search results. Therefore, in order to avoid having to search through tens of pages of irrelevant search results, the inclusion of documents from a particular search engine was stopped after 40 papers in a row met the exclusion criteria.

Backwards and forward search
For all the relevant papers that were found via the aforementioned search queries and passed the filtering stage, all forward and backward references of these papers were examined. All papers were then also included for further examination unless they met the exclusion criteria defined previously.

3.2. Results

In total, this strategy resulted in 21 relevant documents. Five of them did not place a special focus on consolidation but were rather more generally concerned with formalizing legal documents. Note that due to the exclusion criteria, they are still relevant to regulations and would need to at least mention the formalization of modificatory provisions. Additionally, two papers were categorized as being concerned with pre-consolidation topics. Both of

these papers proposed a system, in which the author of a modificatory provision would modify the targetted legal text as opposed to writing the instruction, with the system, in turn, generating this instruction for them. Because this topic is concerned with the author of modificatory provisions and not the user/reader of them, these two papers were not considered further.



The remaining 14 papers were directly concerned with the automation (or partial automation) of the consolidation process. Noteworthy is that none of these papers are considering legal texts in the English language. Rather, they largely stem from Italian researchers, working with Italian regulations with the rest of them concerning themselves with Greek (three papers), Japanese, Spanish, and Polish (one each) regulations. Accordingly, their data sets as well as some NLP techniques are specific to that language and may be only partially reusable for the purposes of this thesis.

The more significant papers from this literature review are described further in the following.

3.2.1. General Formalization of Regulatory Documents

Legal Text Analysis of the Modification Provisions: a Pattern Oriented Approach (Brighi & Palmirani, 2009) In this paper, Brighi & Palmirani propose a methodology for modeling modificatory provisions. In it, they present the following modificatory provision properties

with which to model, classify and annotate modificatory provisions.

- **ActiveNorm (id)** - the provision that states the modification.
- **PassiveNorm (id, internal/external, complete/incomplete, negative/positive, single/-multiple)** - the provision that is affected by the modification. The PassiveNorm can be multiple, incomplete, or expressed with a negative sentence (e.g. repeal all the chapters except the first one).
- **Action (type, duration, date_application, implicit/explicit)** - the action produced by the active provision on the passive provision. The actions are organized in a taxonomy, each action has a date of application including retroactivity and postponement phenomena.
- **Times (Enforcement period (start, end), Efficacy (start, end))** - the times are a couple of intervals that indicate the interval of enforcement of the modificatory provision and the interval of efficacy.
- **Content (role)** - this represents the part of the speech that models the old text to replace or repeal in the modified provision. Sometimes the position indicates where the new text should be inserted (e.g. «Insert before the paragraph beginning with 'If 2557 ...', the following paragraph:»).
- **Purview** - the provision is sometimes used to describe the modification of the range of applications, for explaining an exception in the domain or an interpretation specification.
- **Space** - a parameter used to specify a geographical area to which the modification applies (e.g. the art. 4 is applicable only in the (e.g. «This Act does not apply to the Faroe Islands and Greenland»).
- **Conditions (event, space, domain)** - sometimes the norm is conditioned in its efficacy to an event, geographic space, or a class (or domain) of application (e.g. «suspension of art. 5 for earthquake people of Abruzzo since November 2009»).
- **Reflexivity** - when the ActiveNorm and the PassiveNorm collapse in the same document we have a reflexive modificatory provision, with some side effects on the language.

They argue that this metadata provides all the necessary information "for managing semi-automatically the consolidation process". An evaluation of the expressivity of this model was not performed in this paper.
[BP09]

Model Regularity of Legal Language in Active Modifications (Palmirani & Brighi, 2010) In this paper, Palmirani at el. present a detailed methodology for the classification and detection of modificatory provisions through the use of semantic and syntactic NLP techniques. In this approach, they represent a modificatory provision using the following layers of analysis:

- **TEXT** The part of the document that is officially approved by an authority with legal power.

- **STRUCTURE OF THE TEXT** The part of the document that states a text's organization.
- **METADATA** Any information that was not issued by an authority in its deliberative act. Metadata can involve document description metadata (e.g., keyword), workflow (e.g., procedural steps in the bill), document lifecycle (e.g., document history), and document identification metadata (e.g., URL, URI, URN, and annexes).
- **ONTOLOGY** Any information about the setting in which the document plays a role, for example, information specifying a concept pertaining to the legal system or any concept which is invoked in the text and which needs modeling.
- **LEGAL KNOWLEDGE MODELLING** The interpretation and modeling of the text's legal meaning, especially as concerns the representation of norms and rules that are not already included in the more abstract ontology layer.

This paper mainly serves as an exploratory project, summarizing findings about their data set of 29,000 Italian normative acts. An evaluation of their approach was not performed.

[PB10]

3.2.2. Automatic Consolidatory Metadata Annotation

NLP-based Extraction of Modificatory Provisions Semantics (Mazzei, Radicioni, & Brighi 2009) Based on the research of Brighi & Palmirani, described in 3.2.1, which sets forth a system that aims to model all modificatory provisions, Mazzei et al. aim to annotate legal documents automatically, based on the proposed classifications, by leveraging rule-based parsers and interpreters. Interestingly, they omit the provision properties *Reflexivity*. The paper relies on the input documents to be annotated already being in the *NormaInRete* format, further described in 2.3.1. During the evaluation of their system, they tested the correctness of the following annotations using a hand-annotated ground truth data set:

- **Type** - which can be one of the following: integration, substitution, or deletion
- **Position** - describing the target document as well as the position of the target text in that document
- **Novella** - describing the modifying text
- **Novellando** - describing the text to be modified

In the end, Mazzei et al. were able to measure 82.2% precision and 67.5% recall.
[MRB09]

NLP-based metadata extraction for legal text consolidation (Spinosa et al., 2009 Spinosa et al. propose a system for automatically annotating NIR-based documents (2.3.1) with the necessary metadata to facilitate eventual automatic consolidation. The paper focuses on the semantic analysis of modificatory provisions and more specifically, modificatory

provisions with the following typology of textual amendments: repeal, substitution, and integration.

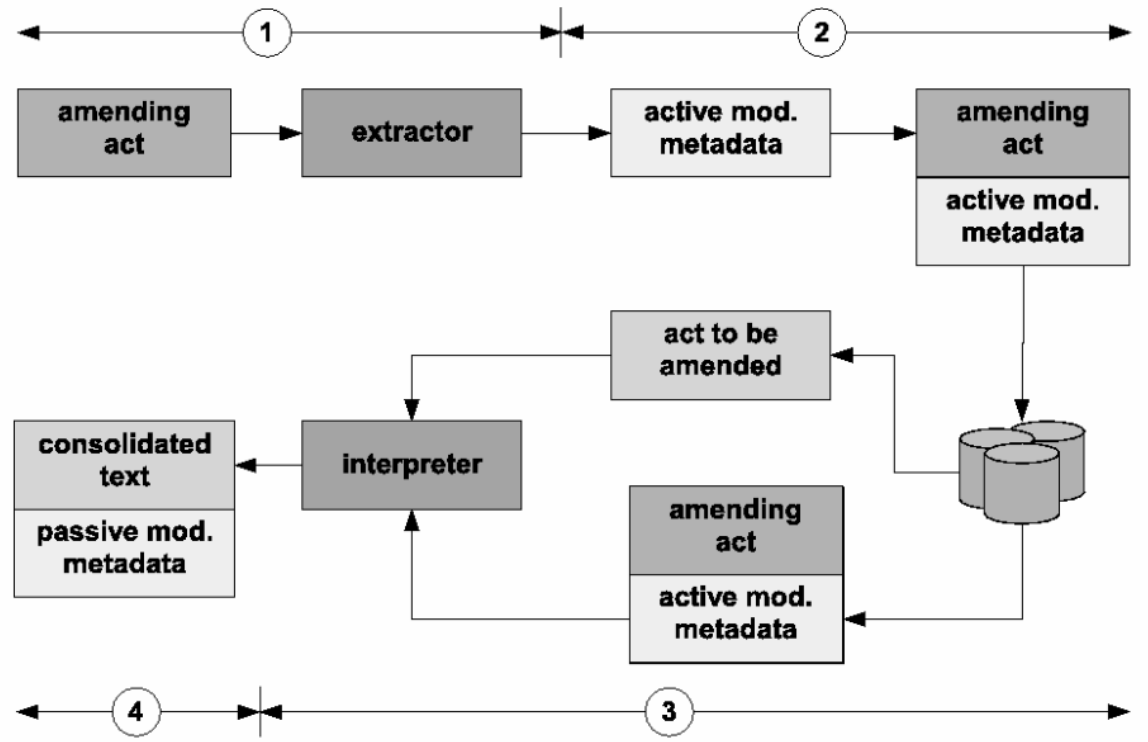


Figure 3.1.: System for automatic consolidation as proposed by Spinosa et al. In this paper, Spinosa et al. only research steps 1 and 2.

Even though technically different, the types of metadata that are being automatically annotated and evaluated match those used by Mazzei et al. [3.2.2] closely. Mainly they are concerned with the type of operation being performed and the location of both the content being targetted as well as the new content being inserted.

Metadata	Description
pos	information on the amending provision
pos:xlink	ID reference to the amending provision
norm	information on the norm to be amended
norm:xlink	URN reference to the norm
– pos	further information on the norm
– pos:xlink	URN reference to the norm with the partition ID
– – border	information on further narrower container
– – border:type	container type (e.g. point, “alinea”, period, etc.)
– – border:num	container label expressed by a number or a letter
– – border:ord	container position expressed by an ordinal (e.g. 2nd) or a relative (e.g. last) number
position	information on the specific modifying point within the narrowest container
– pos	information on a string (quoted) and a bound of the deleting or inserting point
– pos:xlink	ID reference to the string, a bound of which is the beginning of the modifying text
– pos:where	specific bound of the string or container (before, after, start, end)
novellando	information on the outgoing text
– type	information on the “novellando” type
– type:value	“novellando” type (e.g. article, paragraph, “alinea”, period, words, etc.)
– pos	information on the outgoing string (in quotes)
– pos:xlink	ID reference to the string that is either the outgoing text, or the beginning or ending of the outgoing text
– – role	information on the meaning of the string
– – role:value	string role: beginning (from) or ending (up to) of the outgoing text
novella	information on the incoming text
– type	information on the “novella” type
– type:value	novella type (e.g. article, paragraph, “alinea”, period, words, etc.)
– pos	information on the incoming string (quoted)
– pos:xlink	ID reference to the incoming string

Figure 3.2.: Metadata descriptions used for annotation.

During the evaluation of their system, Spinosa et al. were able to record an average of 99.3% precision and 94.8% recall across all types of metadata, with the only values below 90% being the recall of *Position:where* and *Position:pos* with 88.5% and 87.5% respectively. Overall these numbers warrant a good level of confidence in the correctness of the annotations.
[Spi+09]

TULSI: an NLP system for extracting legal modificatory provisions (Lesmo, Mazzei, Palmirani, & Radicioni, 2013) Similar to the previous two research projects [3.2.2][3.2.2], Lesmo et al. set out to propose, implement and evaluate a system that is capable of automatically annotating legal modificatory provisions with the necessary metadata to facilitate full automatic consolidation. For evaluation purposes, Lesmo et al. have decided to use the exact same types of metadata to annotate and evaluate as the ones that Mazzei et al. used in their attempt at automatic metadata annotation of modificatory provisions [3.2.2]. Using a hybrid approach, coupling deep syntactic parsing and shallow semantic interpretation, they were able to achieve an overall precision of 93.54% and 82.00% recall.
[Les+13]

Modicatory Provisions Detection: a Hybrid NLP Approach (Gianfelice, Lesmo, Palmirani, Perlo, Radicioni, 2013 Earlier works regarding automatic annotation of modificatory provisions by Italian researchers [3.2.2][3.2.2][3.2.2] were focussed mainly on modificatory provisions of type *substitution*, *integration* and *repeal*. In contrast to this, this paper is focused on expanding this system to include twelve more modification kinds. These modifications are all concerned in some way with the temporal nature of regulations and their efficacy. For conciseness reasons, however, the paper only fully discusses modifications with the following types:

- *Suspension* - a provision that specifies a time interval, during which an otherwise applicable target provision does not apply
- *Postponement of Efficacy* - a provision that postpones the efficacy of its target, usually an entire document or simple fragments
- *Prorogation of Efficacy* - a provision stating an extension of efficacy for its target provision
- *Exception/Derogation* - a provision stating circumstances under which the target provision does not apply

	accuracy	recall	precision
Postponement	75%	64%	51%
Prorogation	56%	72%	32%
Derogation	68%	63%	46%
Suspension	59%	62%	54%

Figure 3.3.: Breakdown of recall and precision categorized by modification type.

In the end, Gianfelice et al. were able to measure 47% precision and 61% recall on the four modification types that were more deeply examined. In their discussion of results and errors, they list some interesting examples of wrong input data and edge cases.

Automatic Extraction of Amendments from Polish Statutory Law (Smywiński-Pohl et al., 2021 This research project by Smywiński-Pohl et al. aims to leverage recent neural network models such as BERT and BiRNN in order to not only detect amending provisions but also to classify them and extract metadata that is potentially needed for automatic consolidation.

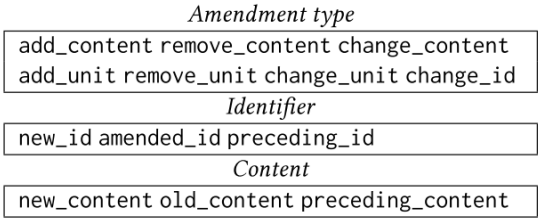


Figure 3.4.: Types of metadata and their possible values annotated in the amendments.

The detection rate recorded in the evaluation of their implementations seems to be highly encouraging, with 4/5 models tested achieving f1 scores well above 95%.

Model	Micro F1	Macro F1	Weighted F1	Support
Rules	69.58	82.03	72.07	1335
HerBERT	97.96	90.81	97.92	1174
RoBERTa	97.69	97.68	97.70	1148
XLMR	96.72	84.62	96.73	1174
BiRNN 1	90.81	80.72	90.60	1773
BiRNN 2	**98.20**	**98.90**	**98.19**	1174

Figure 3.5.: Results of amendment detection and classification experiment.

Note that these f1 scores are not only representative of amendment detection but also the correct annotation of content and relevant identifiers. However, the authors mention that there are certain pre-processing steps being performed in order to achieve these scores which are specific to the structure of this particular data set, consisting of 242 bills of Polish statutory law, hurting the generalizability of their approach.

3.2.3. Computer-Assisted Legal Consolidation

Relationship-based dynamic versioning of evolving legal documents (Martínez et al. 2003) To combat the discrepancy between the way document authors perform amendments on regulatory documents and the way document users want to view different document versions, Martinez et al. propose a relationship graph-based solution, using XLink. In this approach, references to other documents are modeled as links with which documents can then be placed into a relationship graph. This graph can then be traversed in order to dynamically create the desired document version. This approach, in theory, presents a way of completely automating the consolidation process, provided that external reference links, consisting of:

- link target: document tree item to substitute
- link source: document tree item that substitutes

can be correctly and reliably detected.

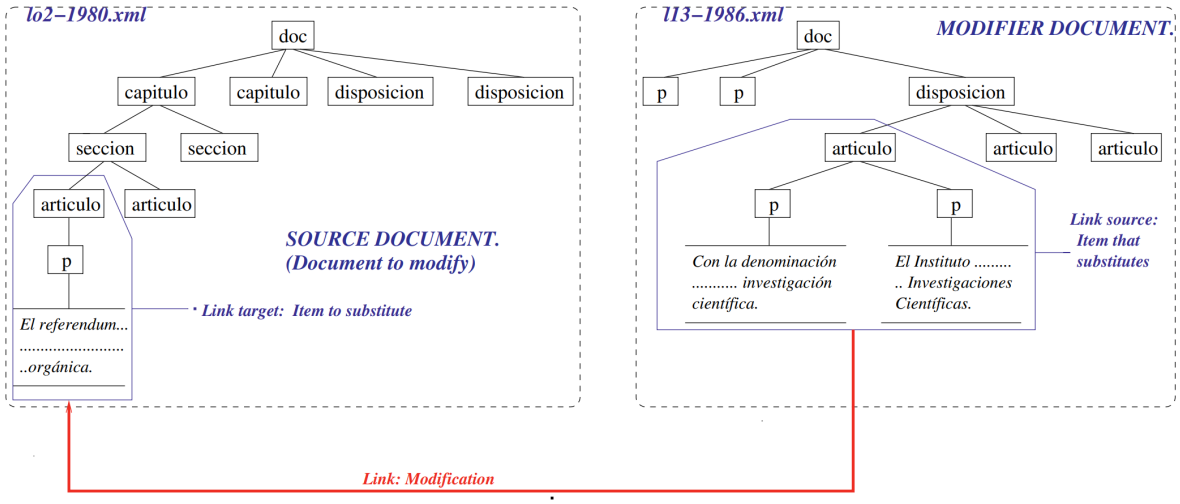


Figure 3.6.: Our model of a document.

During the evaluation of their prototype using this approach, only the detection rate of external links has been tested, stating that in a set of 50 documents, 90% of references were detected correctly. Whether this is referring to the accuracy or recall and how this correctness relates to the detection of link targets and/or link sources was not specified in the paper. The paper further makes no mention of evaluating the correctness of the eventual dynamically generated document version.
[Mar+03]

Automatic Consolidation of Japanese Statutes Based on Formalization of Amendment Sentences (Ogawa et al., 2008) This paper also mentions the process of regulatory document consolidation as its core problem statement. In it, Ogawa et al. develop a classification of amendment clauses in the Japanese language, represented by sixteen regular expressions. They further propose a system for automatic consolidation using this classification, claiming only 23 errors in 1,164 amendment operations. The application of changes and detection of the change target were also performed using regular expressions. The paper points out that operations on tables and amendment sentences themselves were excluded from this result, as they present specific further complications and represent a relatively small percentage of the modificatory provisions (7% for table instructions, 0.6% for amendments on amendment sentences).
Since this approach is based on regular expressions specific to the Japanese language, this

approach does not seem to be transferable to the efforts of this paper, which aims to specify a more general formalization of modificatory provisions.
[OIT08]

A semi-automatic system for the consolidation of Greek legislative texts (Garofalakis, Plessas, Plessas, 2016 Garofalakis et al. present a semi-automatic system for the consolidation of Greek legislative documents based on regular expressions. The initial plain text documents, parsed from PDF, are converted to XML which is validated against a custom schema written for Greek legislation, thereby implementing a quasi-dialect of XML, similar to *Akoma Ntoso* and *NIR* [2.3]. A manual final step is required because of system failures caused by errors in the original text or syntax errors. An evaluation of the accuracy of the performed consolidation operations was not part of the paper.
[XPP16]

3.3. Research Gap

There have been numerous attempts at automating the legal consolidation process. Mainly, this research has been performed by Italian researchers, since a lot of the Italian regulations are published in NIR, a format that is easily machine-readable and extendable with metadata annotations. Generally, most of this research is focused heavily on the annotation and information extraction of amending documents and less so on the automatic application of these changes in order to create the correct consolidated document.

In conclusion, there appears to be a research gap mainly concerning research about regulatory documents in the English language as well as research about the automatic change application of correctly formalized modificatory provisions. This thesis' research questions are formulated accordingly.

4. Data Analysis and Requirements Elicitation

This chapter describes and analyzes the data set of regulatory documents. From it, a set of requirements, needed for amending documents to be formalized into a machine-readable, automatically executable format, is elicited.

4.1. Data Set Description

The data set consists of three document categories:

- Unstructured PDF documents
- Base document versions in HTML
- Hand-consolidated document versions in HTML

4.1.1. Unstructured PDF Documents

A large part of the data set consists of PDF documents for the regulations of the UNECE and US (Code of Federal Regulations Title 49, Part 571), specifically concentrated on the automotive industry. These documents were sampled in order to elicit the requirements for the formalization format for modificatory provisions as well as the consolidation engine reference.

4.1.2. Certivity Document Representation

There are three main data structures with which Certivity has modeled regulatory documents:

1. **Documents**
   Contains metadata about documents
2. **Composition Objects**
   Contains textual content representation in HTML format.
   - *html*: String field, containing textual content in HTML. Always surrounded by tag.

3. **Document Statements**
   Denotes hierarchical position of composition object in the document.
   - *sortValue*: Increasing number, representing the relative position in the document. Lower sortValue → earlier in the document and vice versa.
   - *parent*: Reference to parent document statement.
   - *document*: Reference to the document it is part of.
   - *type*:
     - *Creates*: Denotes the first time a chapter has been introduced. This can simply take place in a base document version or the chapter is retroactively added through an amending document.
     - *Deletes*: The chapter is deleted in this document version.
     - *Amends*: The chapter's content has been changed in this version.
     - *Duplicate*: The chapter is mentioned in this amending document but the content remains the same.
     - *Silent Duplicate*: This chapter has not been mentioned in the amending document and is supposed to be silently copied without changes from the previous version.

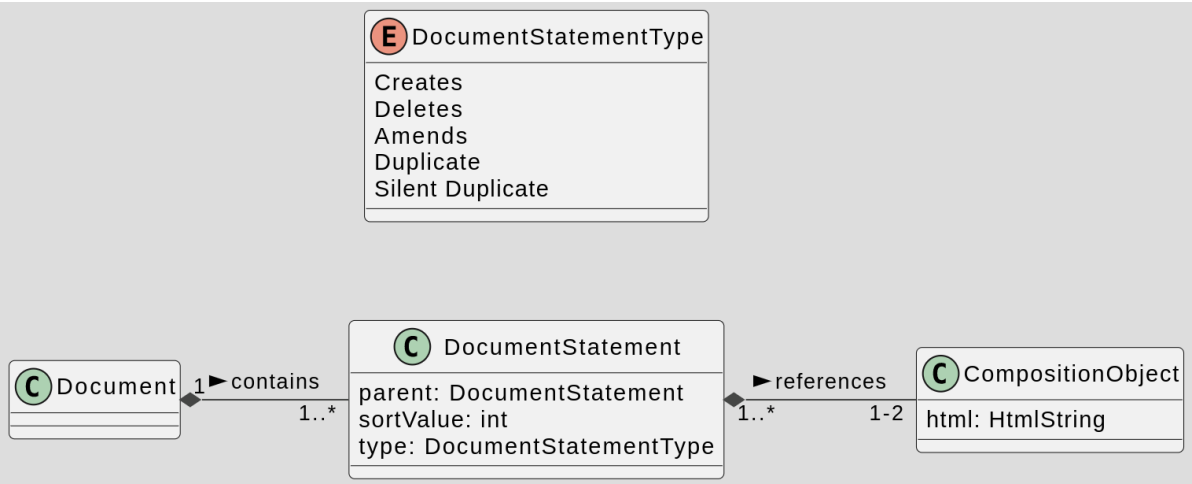


Figure 4.1.: Certivity document representation.

4.1.3. Base Document Representations & Consolidated Documents

Base document and consolidated document versions are converted manually by regulatory experts at Certivity to the Certivity document representation format mentioned in chapter [4.1.2].

4.1.4. Modificatory Provisions

UN data is available as HTML, hand-parsed by regulatory experts at Certivity. US data is parsed from PDF strictly for the purposes of this thesis.

	# amending documents	# modificatory provisions
UN	291	2173
US	74	1493

4.2. Analysis and Classification of Modificatory Provisions

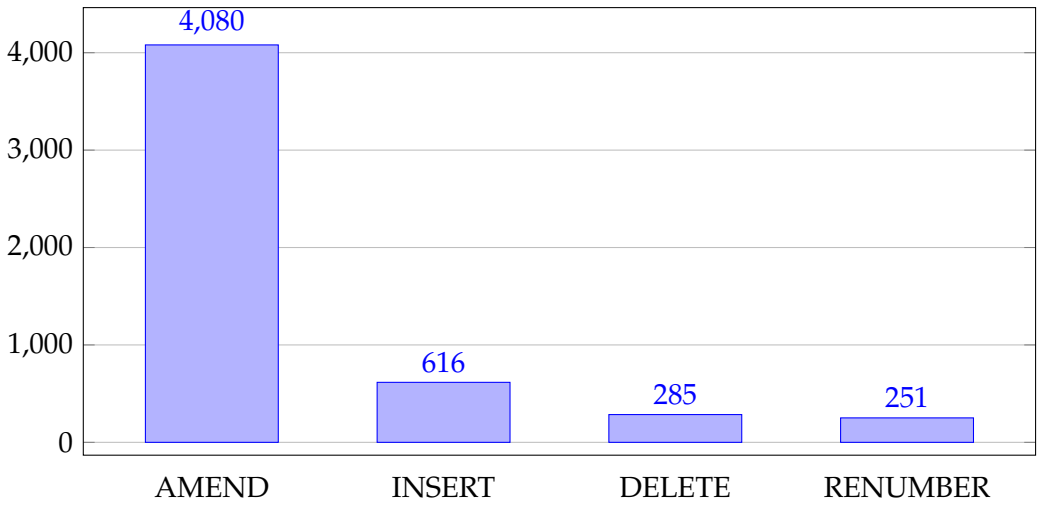
The following keywords were detected in the data set and were categorized in the following way:

DELETE instruction keywords
Found Examples: *"delete"*, *"remove"*
Occurences in the data set: 285 times

RENUMBER instruction keywords
Found Examples: "*renumber*", "*re-number*", "*redesignate*"
Occurences in the data set: 251 times

INSERT instruction keywords
Found Examples: *"introduce"*, *"insert"*, *"add"*
Occurences in the data set: 616 times

AMEND instruction keywords
Found Examples: *"amend"*, *"replace"*, *"revise"*, *"restructure"*, *"correct to read"*, *"continues to read"*
Occurences in the data set: 4080 times



In both the UN and US regulations, amending instructions that fall under the AMEND category are by far the most common. Although the difference among the remaining categories is not as large in comparison to the AMEND category,

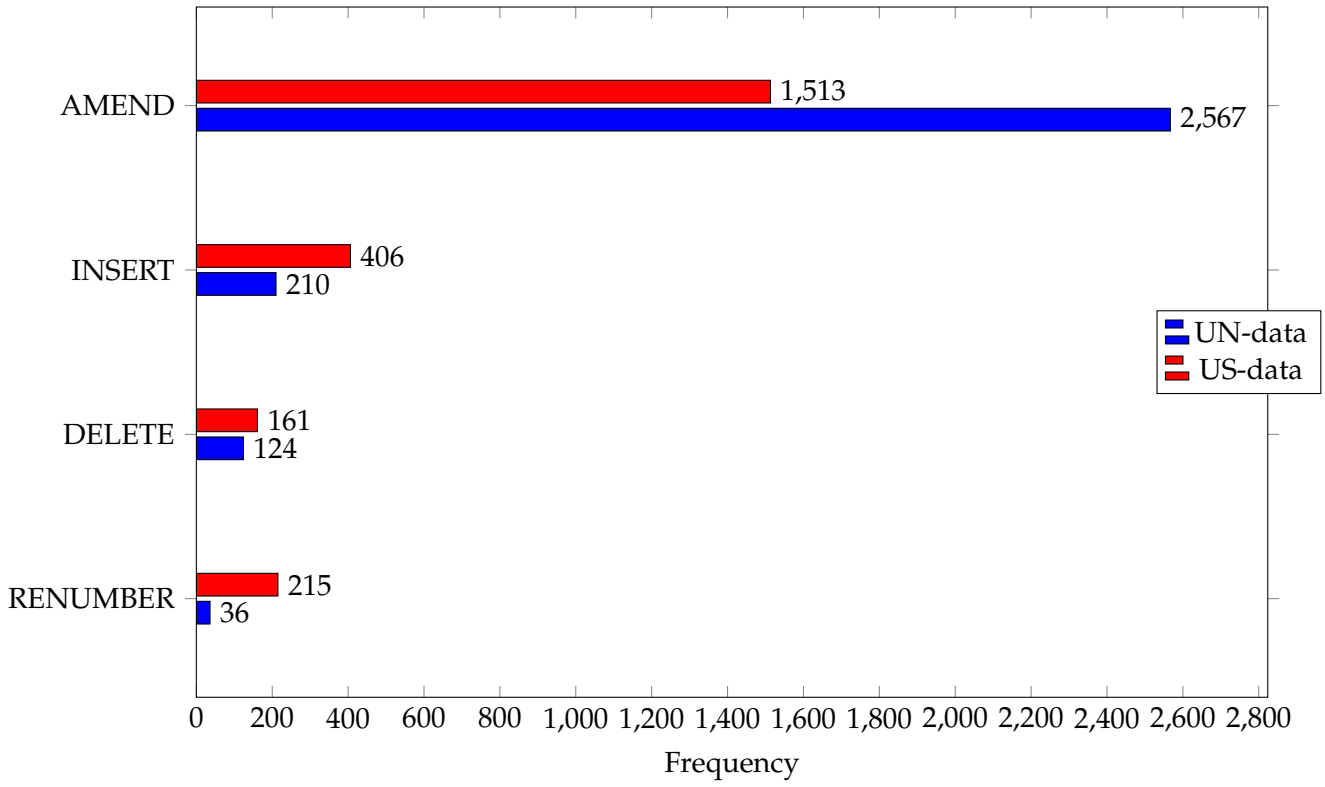


Figure 4.2.: Amending instruction type frequency across the two regions.

It is also noteworthy that at least in the UN-data, the frequency of amendments as well as the number of modificatory provisions have increased over time as shown in the following graph.

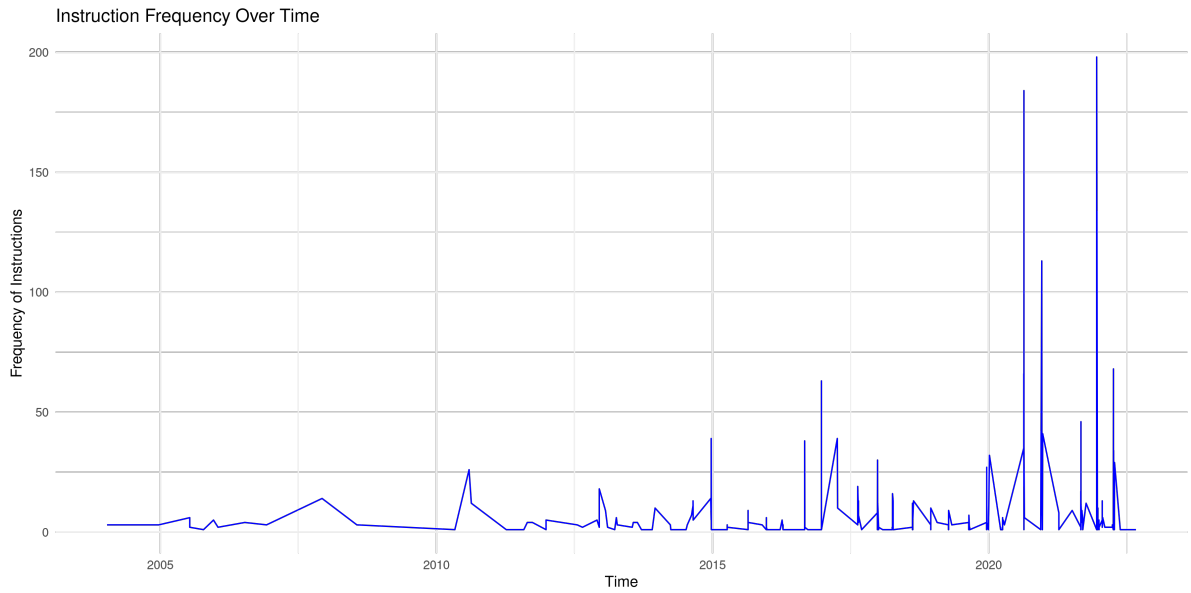


Figure 4.3.: Frequency and amount of amending instructions from regulatory updates within the UNECE data set.

4.3. Requirements for Modificatory Provisions Representation

In this section, we describe the parameters that are required for each consolidation engine function elicited in chapter 4.2. All of them assume that a base document, with the necessary annotated metadata (as described in 4.4), has already been provided and is acting as the target document on which these operations are performed.

4.3.1. DELETE Operation Parameters

The DELETE operation is the simplest out of all of the operations, requiring only the target chapter identifier (further explained in chapter 4.4.2) as the sole parameter for this function.

Parameters:

- **Target Chapter Identifier**
  Chapter to be removed.

4.3.2. INSERT Operation

INSERT operations are more complicated than a DELETE operation, however, it only requires one additional parameter: *New Text Content*. This text needs to be in the same representation

format, in which the base document was provided to the consolidation engine, or be easily translatable to the base document representation format.

In order to find the correct position in the tree hierarchy at which to insert this new chapter, we need to be able to derive the correct parent chapter identifier from the *New Chapter Identifier* parameter. This operation is the main reason that chapter identifiers need to have their parent identifier unambiguously derivable from them as further detailed in chapter 4.4.2.

Parameters:

- **New Chapter Identifier**
  Identifier of the newly introduced chapter, parent identifier must be unambiguously derivable from this.
- **New Text Content**
  Text content of newly introduced chapter. Should be already in the same format as the base document representation or be easily translatable.

4.3.3. RENUMBER Operation

A renumber operation can be modeled as a DELETE operation of the target chapter, followed by an INSERT operation, re-inserting the old chapter with a new correct identifier as well as its new corresponding position in the tree hierarchy.

Parameters:

- **Target Chapter Identifier**
  Chapter to be renumbered.
- **New Chapter Identifier**
  Identifier of the newly introduced chapter, parent identifier must be unambiguously derivable from this.

4.3.4. AMEND Operation

An AMEND operation can be performed in multiple ways.

4.3.4.1. REPLACE - replace all

The most straightforward AMEND case can be described as a REPLACE operation, overriding the content of a chapter wholly with new textual content. Similarly to the RENUMBER operation, the REPLACE operation can be expressed as a combination of a DELETE + INSERT operation and as such, the parameters are simply a union of the DELETE and INSERT parameters.

Parameters:

- **Target Chapter Identifier**
  Chapter to be replaced.
- **New Text Content**
  Text content of newly introduced chapter. Should be already in the same format as the base document representation or be easily translatable.

> *Paragraph 15.2.1.2.*, amend to read:
>
> "15.2.1.2. The provisions of this Regulation do not apply to the surveillance mirrors defined in paragraph 2.1.1.3. Nevertheless, the exterior surveillance mirrors shall be mounted at least 2 m above the ground when the vehicle is under a load corresponding to its maximum technical permissible mass or shall be fully integrated in a housing including Class II or III mirror(s) which is (are) type approved to this Regulation."

Figure 4.4.: An example of a REPLACE instruction, taken from a modificatory provision from the UN data. [UNR46/4.3.0/proposed]

4.3.4.2. AMEND - replace partly, using ellipses

] In our data, there are plenty of modificatory provisions, which do not state the whole update paragraph in full. Rather they use ellipses to imply that content from the original paragraph is supposed to be filled in, in an unmodified manner. Unfortunately, this demarcation of change borders is often very implicit and relies heavily on human understanding in order to find the correct text to be modified.

Parameters

- **Target Chapter Identifier**
  Chapter to be amended.
- **New Chapter Content**
  New content that is replacing the old content. This text is supposed to contain ellipses.

To uniquely instruct the consolidation engine on which parts of the original text the amending text should attach, the author needs to provide enough context before and/or after the ellipsis.

Modificatory Provision:

Paragraph 16.13.1., amend to read:

"16.1.3.1. Magnification factor

The minimum and the average magnification factors of the CMS, in both horizontal and vertical directions shall not be lower than the magnification factors indicated below.

……"

Target Text:

16.1.3.1. Magnification factor

The minimum and the average magnification factors of the CMS, in both horizontal and vertical directions shall not be lower than the minimum average magnification factor indicated below.

The minimum magnification factor shall not be less than:

(a) for Class I: 0.31;

(b) for Class II (driver's side): 0.26;

Intended Outcome:

16.1.3.1. Magnification factor
The minimum and the average magnification factors of the CMS, in both horizontal and vertical directions shall not be lower than the ~~minimum average~~ magnification ~~factor~~ factors indicated below.
The minimum magnification factor shall not be less than:
(a) for Class I: 0.31;
(b) for Class II (driver's side): 0.26;

Figure 4.5.: Example of a modificatory provision with an ellipsis at the end of a chapter. [UNR46/4.5.0/proposed]

In the above example, since the ellipsis is at the end of the modificatory provision text, enough context from the original text needs to be provided in order to match the correct text from the original document that is supposed to take place of the ellipsis. In this case, "indicated below." is enough context since it does not appear twice in the text.

Modificatory Provision:

Paragraph 6.2.1.2., amend to read:

"6.2.1.2. If a device for indirect vision …… the total process of scanning, rendering and reset to its initial position together shall not take more than 200 milliseconds at room temperature of 22 °C ± 5 °C."

Target Text:

6.2.1.2. If a device for indirect vision can only render the total prescribed field of vision by scanning the field of vision, the total process of scanning, rendering and reset to its initial position together shall not take more than 2 seconds.

Intended Outcome:

6.2.1.2. If a device for indirect vision can only render the total prescribed field of vision by scanning the field of vision, the total process of scanning, rendering and reset to its initial position together shall not take more than ~~2 seconds~~200 milliseconds at room temperature of 22 °C ± 5 °C.

Figure 4.6.: Example of a modificatory provision with an ellipsis in the middle of a chapter. [UNR46/4.4.0/proposed]

In the above example, since the ellipsis is in the middle of the modificatory provision text, the amending document author needs to provide enough context for the text before AND after the ellipsis. In this case, "If a device for indirect vision" and "the total process of scanning, rendering" are sufficient.

4.3.4.3. AMEND - replace keywords

Sometimes, a modificatory provision will instruct to replace all occurrences of a certain keyword with another. This instruction may be applied to all child chapters as well.

Parameters

- **Target Chapter Identifier**
  Chapter to be amended.
- **Keyword**
  Keyword to be replaced.
- **Replacement**
  Word which gets inserted in the place of the aforementioned keyword.
- **Recursive**
  States whether this operation is to be applied recursively to all child chapters or not.

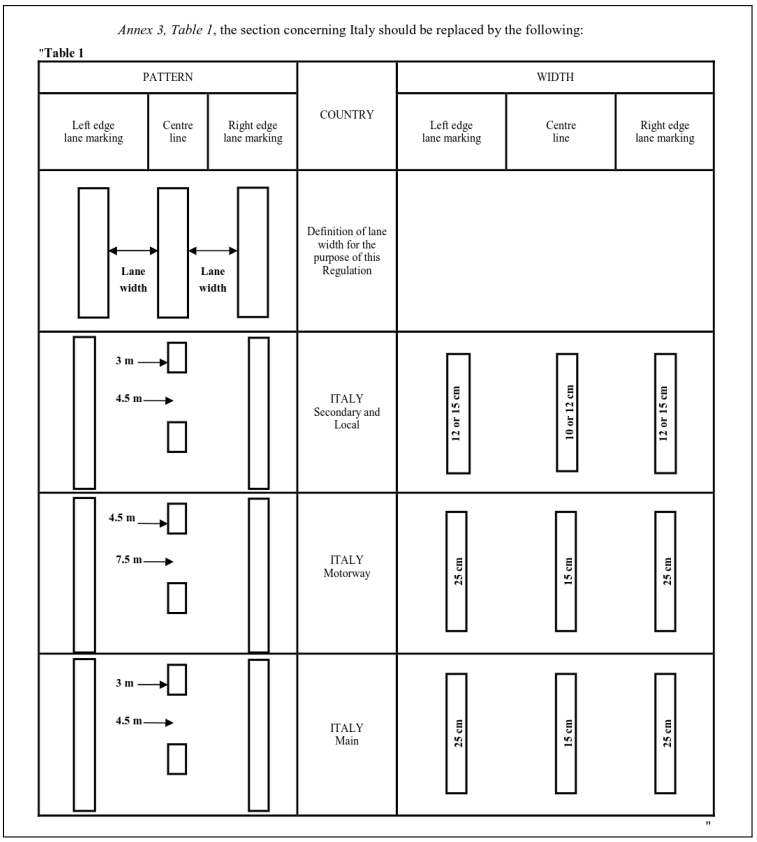
4.3.5. Targetting non-chapter entities

Targeting non-chapter entities (defined more closely in chapter 2.1.10) presents a special set of challenges since they do not follow the same numbering scheme that chapters do. Since these document sections can still be the target of modificatory provisions, requirements for the formalization of modificatory provisions targetting certain non-chapter entities will be defined in the following subchapters. For the sake of brevity, only the requirements for the formalization of the basic INSERT, DELETE, and REPLACE instructions will be further detailled. More complicated instructions like REPLACE_PARTLY or REPLACE_KEYWORD will be ignored.

4.3.5.1. Tables

Amending tables represent a special edge case in which only certain parts of a table are being targeted for amendments. In these cases, it is particularly challenging to encode which part of the table is to be amended, since the targeting used in amending documents is often context based.

Modificatory Provision:



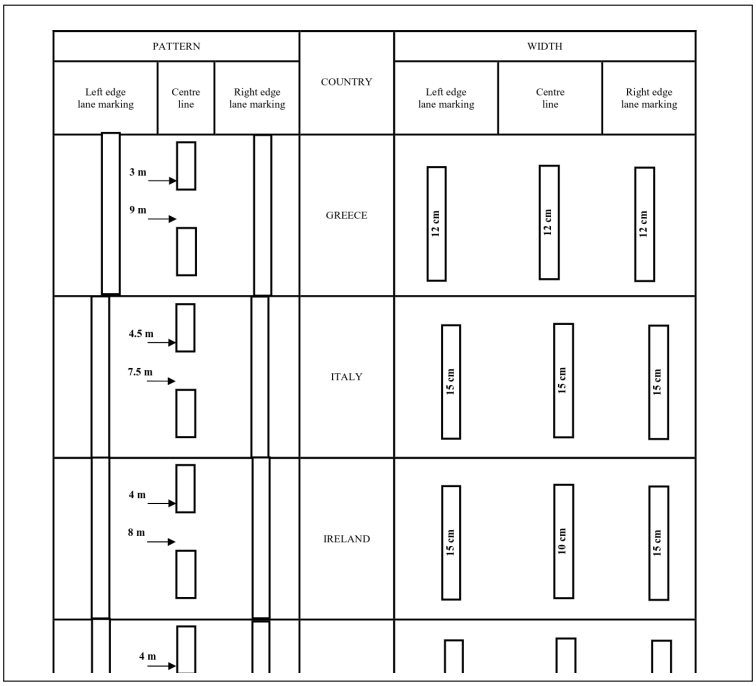
Annex 3, Table 1, the section concerning Italy should be replaced by the following:

"**Table 1**

PATTERN			COUNTRY	WIDTH		
Left edge lane marking	Centre line	Right edge lane marking		Left edge lane marking	Centre line	Right edge lane marking
Lane width **Lane width**			Definition of lane width for the purpose of this Regulation			
3 m, 4.5 m			ITALY Secondary and Local	12 or 15 cm	10 or 12 cm	12 or 15 cm
4.5 m, 7.5 m			ITALY Motorway	25 cm	15 cm	25 cm
3 m, 4.5 m			ITALY Main	25 cm	15 cm	25 cm

"

Target Text:



PATTERN			COUNTRY	WIDTH		
Left edge lane marking	Centre line	Right edge lane marking		Left edge lane marking	Centre line	Right edge lane marking
3 m, 9 m			GREECE	12 cm	12 cm	12 cm
4.5 m, 7.5 m			ITALY	15 cm	15 cm	15 cm
4 m, 8 m			IRELAND	15 cm	10 cm	15 cm
4 m						

Figure 4.7.: Example of a context specific AMEND_TABLE instruction.

However, accurate targetting is still possible as long as every group of rows to be targetted has the necessary metadata annotation that can be used for the instruction. These row groupings can then be modified with the following operations:

Parameters - INSERT

- **Table ID**
  Target table ID, e.g. *Annex 3, Table 1*
- **New Rows**
  New Rows, which are to be inserted

Parameters - DELETE

- **Table ID**
  Target table ID, e.g. *Annex 3, Table 1*
- **Table Group ID | Row Numbers**
  Targets the table group to be deleted. Alternatively, rows could be targetted via their row number although this might make the operation flaky depending on the operations that were applied previously to the table.

Parameters - REPLACE

- **Table ID**
  Target table ID, e.g. *Annex 3, Table 1*
- **Table Group ID | Row Numbers**
  Targets the table group to be deleted. Alternatively, rows could be targetted via their row number although this might make the operation flaky depending on the operations that were applied previously to the table.
- **New Rows**
  New Rows, which are to be inserted

4.3.5.2. Footnotes

Since footnote anchors might reset between chapters, it is important to provide both the target chapter as well as the anchor value of the footnote to be targeted.

Parameters - INSERT

- **Target Chapter Identifiers**
  List of chapters that this footnote is referenced in
- **Anchor**
  Anchor value of the footnote
- **Content**
  Content of the footnote

Inserting footnotes like this will need to be followed up with AMENDS instructions, adding in the footnote anchors for each of the affected chapters.

Parameters - DELETE

- **Target Chapter Identifiers**
- **Anchor**

This DELETE instruction will only delete the footnote in case it is no longer referenced in the entire document. Otherwise, it will simply remove the reference from the affected chapters.

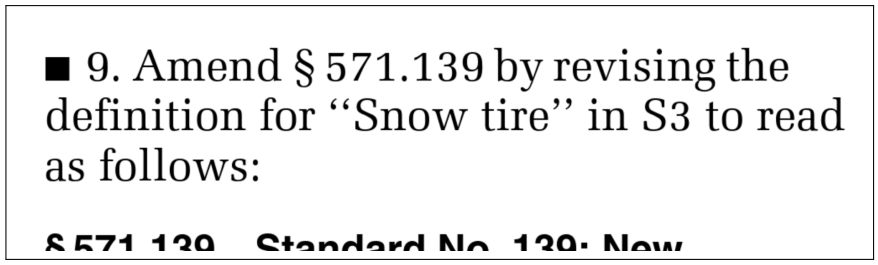
Parameters - REPLACE

- **Target Chapter Identifier**
- **Anchor**
- **New Content**

4.3.5.3. Definition Items

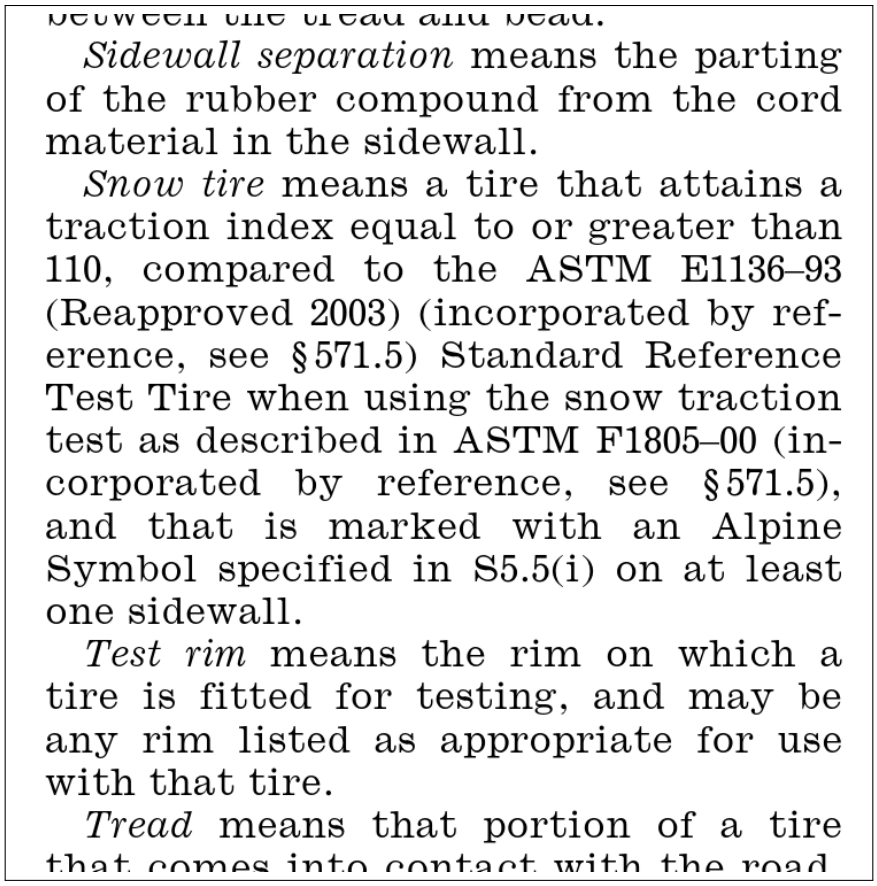
In the US data set, there exist a lot of chapters that are comprised of a list of terms with their respective definition. Amending documents will often instruct to change the definition of a certain term specifically.

Modificatory Provision:


■ 9. Amend § 571.139 by revising the definition for ''Snow tire'' in S3 to read as follows:

§ 571.139 Standard No. 139; New

Target Text:


between the tread and bead.

Sidewall separation means the parting of the rubber compound from the cord material in the sidewall.

Snow tire means a tire that attains a traction index equal to or greater than 110, compared to the ASTM E1136–93 (Reapproved 2003) (incorporated by reference, see §571.5) Standard Reference Test Tire when using the snow traction test as described in ASTM F1805–00 (incorporated by reference, see §571.5), and that is marked with an Alpine Symbol specified in S5.5(i) on at least one sidewall.

Test rim means the rim on which a tire is fitted for testing, and may be any rim listed as appropriate for use with that tire.

Tread means that portion of a tire that comes into contact with the road

Figure 4.8.: Example of definition item targeting. [UNR12 4.4.0 enforced]

Parameters - INSERT

- **Target Chapter Identifier**
- **Keyword**
  Keyword to be defined
- **Definition**

Definition of the above keyword

Parameters - DELETE

- **Target Chapter Identifier**
- **Keyword**

Parameters - REPLACE

- **Target Chapter Identifier**
- **Keyword**
  Keyword to be re-defined
- **Definition**
  New definition of the above keyword

4.4. Requirements for Base Document Representation

The requirements for a base document representation are entirely dependent on the requirements of the modificatory provisions formalization. More specifically, almost all requirements have to do, in some way, with enabling the targetting of every entity that might get referenced for changes in a modificatory provision.

4.4.1. Unique Document Identifier

All documents (including consolidated documents) need to have a unique identifier, specifying the regulation it belongs to as well as the current version it represents.

4.4.2. Unique Chapter Identifiers

All modificatory provision categories examined in the previous chapter [4.3] require a unique chapter identifier to specify the target text which is to be modified. In any base document version, all chapters which are later referenced in a modificatory provision, therefore, need to be annotated with an identifier that is compliant with the following requirements in order to reliably find the relevant text to be modified and perform any operations on it.

Document-Level Uniqueness: All chapters within a base document should be annotated with an identifier that is unique on a per-document basis. They are not required to be unique across different documents or document versions.

Uniquely Derivable Parent Identifier: From any identifier, its parent identifier should be clearly and uniquely derivable. As an example, the parent identifier of *"Annex 4 Chapter 2.5.4."*

is *"Annex 4 Chapter 2.5."*. We define a special identifier *"Root"* as the parent of all chapters that have no chapter parent, i.e. this thesis' chapter with the identifier *"1. Introduction"* is *"Root"*. *"Root"* therefore is the only identifier from which no parent identifier can be derived.

Examples:

Annex 4 Chapter 2.5.4.	$\xrightarrow[\text{parentId}]{\text{derive}}$	*Annex 4 Chapter 2.5.*
1. Introduction	$\xrightarrow[\text{parentId}]{\text{derive}}$	*Root*
Root	$\xrightarrow[\text{parentId}]{\text{derive}}$	*null*

Uniquely Derivable Younger Sibling Identifier: From any chapter identifier, the younger sibling must be uniquely derivable.

Examples:

Annex 4 Chapter 2.5.4.	$\xrightarrow[\text{sibling}]{\text{younger}}$	*Annex 4 Chapter 2.5.3.*
2. Fundamentals	$\xrightarrow[\text{sibling}]{\text{younger}}$	*1. Introduction*
Annex 4 Chapter 2.5.1.	$\xrightarrow[\text{sibling}]{\text{younger}}$	*null*

4.4.3. Hierarchical Tree Structure

The hierarchical tree structure is vital for a lot of modificatory operations. It enables:

- **Iterating child chapters**: e.g. during a REPLACE_KEYWORD operation, which is stated to be performed recursively, we need to be able to iterate all children.
- **Treating a node with children as one**: e.g. during a DELETE operation with target chapter 2.1., we want to also delete every child node with it unless otherwise stated.
- etc.

4.4.4. Labelling of non-Chapter Entities

As discussed in chapter 2.1.10, non-chapter entities do not follow the usual numbering scheme for chapters and their position in the document

4.4.4.1. Tables

Each table should have a unique identifier by which they can be targeted.

Table Rows Usually, tables tend to have a column, which acts as the key for each row (i.e. every entry in this column is distinct from one another). By stating this column name in the metadata of the table object, it can be used to target specific rows.

Table Sections Table sections that belong together on a semantic level and can be targeted need to be grouped together. As an example in HTML, this could be done via a <trbody> tag. This is required for certain amend operations as discussed in 4.3.5.1.

4.4.4.2. Images

Each image should have an identifier. This identifier can optionally be unique on a global level to enable even easier targeting but chapter-based uniqueness is sufficient.

4.4.4.3. Footnotes

Footnotes should be denoted as such with a special tag. Additionally, all footnotes in our data set have a mandatory anchor with which they can be matched to the corresponding relevant text passages. This anchor should be clearly denoted in the metadata of the corresponding footnote element. In addition to this, every other chapter entity should also be annotated with a list containing every footnote anchor referenced in its text.

Some examples of these footnote designations include:

Superscript Numbered Anchors:

1.1. This Regulation applies to the braking of vehicles of categories M_1 and N_1.[1]

[1]This Regulation offers an alternative set of requirements for category N1 vehicles to those contained in Regulation No. 13. Contracting Parties that apply both Regulation No. 13 and this Regulation recognize approvals to either Regulation as equally valid. M1 and N1 categories of vehicles are defined in the Consolidated Resolution on the Construction of Vehicles (R.E.3.), document ECE/TRANS/WP.29/78/Rev.3, para. 2 - www.unece.org/trans/main/wp29/wp29wgs/wp29gen/wp29resolutions.html

Incremented Star Anchors:

This Regulation covers new pneumatic tyres designed primarily for vehicles of categories M_1, N_1, O_1 and O_2. */ **/
It does not apply to tyres designed primarily for
(a) the equipment for vintage cars
(b) competitions.

*/ As defined in Annex 7 to the Consolidated Resolution on the Construction of Vehicles R.E.3 (document TRANS/WP.29/78/Rev.1 as last amended by Amend. 4).
**/ This Regulation defines requirements for tyres as a component. It does not limit their installation on any categories of vehicles.

Underlined Numbered Anchors:

2.6. "Bead" means the part of a pneumatic tyre which is of such shape and structure as to fit the rim and hold the tyre on it; 2/

2/See explanatory figure.
Explanatory figure
(see paragraph 2 of the Regulation)

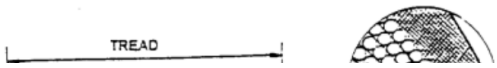


Figure 4.9.: [UNR13H 1.0.0 & UNR30 2.17.1]

These anchors are not guaranteed to be unique on a document basis. Rather they are usually only unique on a page or chapter basis, resetting between the respective entities. E.g. one might find footnotes 1-3 on page 34 but then another footnote on page 36, which also has 1 as its anchor value.

4.4.4.4. Definition Items

In the UN data set, each definition tends to have its own chapter numbering with which they can be targeted. In contrast to this, definitions in the US regulatory documents usually are all lumped together in a chapter together, making it harder to target a specific term definition for amendments.

UN definitions with chapter numbering:

2.	DEFINITIONS
	For the purposes of this Regulation,
2.1.	"Type of pneumatic tyre" means a category of pneumatic tyres which do not differ in such essential respects as:
2.1.1.	The manufacturer;
2.1.2.	Tyre-size designation;
2.1.3.	Category of use (ordinary (road-type) or snow tyre or for temporary use);
2.1.4.	Structure (diagonal (bias-ply), bias-belted, radial-ply, run flat tyre);

US definitions without identifiers:

S4. *Definitions.*

Ackerman Steer Angle means the angle whose tangent is the wheelbase divided by the radius of the turn at a very low speed.

Drive configuration means the driver-selected, or default, condition for distributing power from the engine to the drive wheels (examples include, but are not limited to, 2-wheel drive, front-wheel drive, rear-wheel drive, all-wheel drive, 4-wheel drive high gear with locked differential, and 4-wheel drive low gear).

Electronic stability control system or *ESC system* means a system that has all of the following attributes:

To facilitate the targetting of specific definition terms, each one of the definition items would need to get annotated as such. Since the term to be defined is always unique on a chapter basis, this value can be used as a key.

5. Reference Implementation

5.1. Conclusion and HTML as Representation Format

There are three main factors determining which document representation format is best suited as the basis for the consolidation engine to perform its operations:

1. **Requirements elicited in chapter 4.4**
   Mainly, those are:
   - Hierarchical tree structure
   - Unique chapter identifiers
2. **Compatibility with existing legal document representation formats [2.3]**
   Existing representation formats are almost exclusively based on XML
3. **Availability of test data**
   Especially consolidated documents are rarely available in formats other than PDF.

In accordance with these three criteria, we have decided to go with HTML as our base document representation format. It fulfills all the necessary requirements we have elicited in chapter 4.3 and is easily translatable into any XML dialect. Arguably the most important factor for this decision is the fact that the representation format employed by Certivity uses HTML to encode textual content [4.1.2]. As this thesis is written in collaboration with Certivity, it presents the best way to obtain consolidated documents in the English language in a structured format and will help greatly with automation during the evaluation step of this thesis.

5.1.1. Transformation of Certivity data into HTML

Before we can feed the structured data provided by Certivity [4.2] to the consolidation engine, the data first needs to be converted into pure HTML [5.1]. This can be achieved by nesting the HTML content of child composition objects into their parent's HTML content, as visualized in figure 5.1.

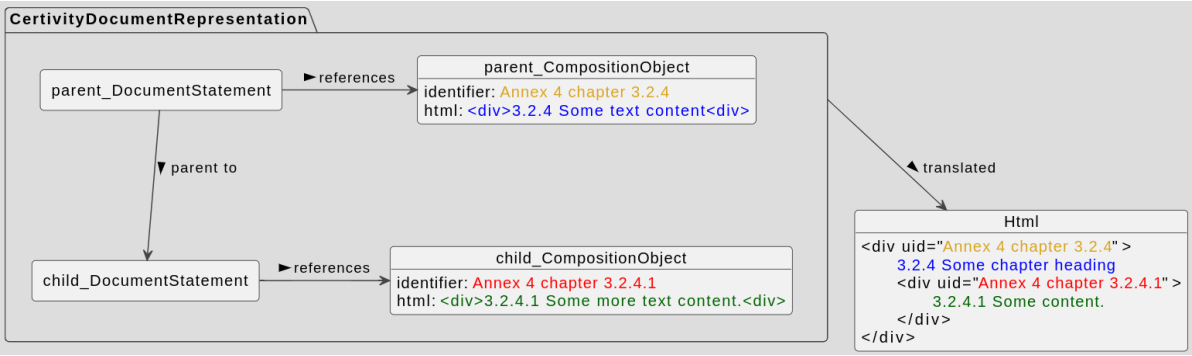


Figure 5.1.: Transformation of Certivity data structure into pure HTML.

5.1.2. Consolidation Engine

The consolidation engine was written in TypeScript. TypeScript was chosen, as it provides a number of benefits specific to this use case:

- *Good DOM API*: This simplifies the large amounts of DOM manipulation operations that need to be performed on the HTML representation of the base document.
- *Easy embedding into different UIs with the web platform*: TypeScript UIs are able to run in a platform-independent way in a web browser or as desktop apps with the help of technologies such as Electron and Tauri.
- *Ease of library publishing*: NPM facilitates easy writing, publishing, and importing of libraries.
- *Popularity*: JavaScript and by extension TypeScript are among the most commonly used programming languages in the world. Software developers wanting to contribute or build on top of the consolidation engine will likely know how to code in JavaScript.

The consolidation engine is published on the NPM platform (https://www.npmjs.com/package/legal-consolidator) and as a result, is able to be imported into any Node program.

Its API exposes the following public functions and types, visualized in the UML diagram in figure 5.2 and described further in subsequent chapters 5.1.3 to 5.1.6.

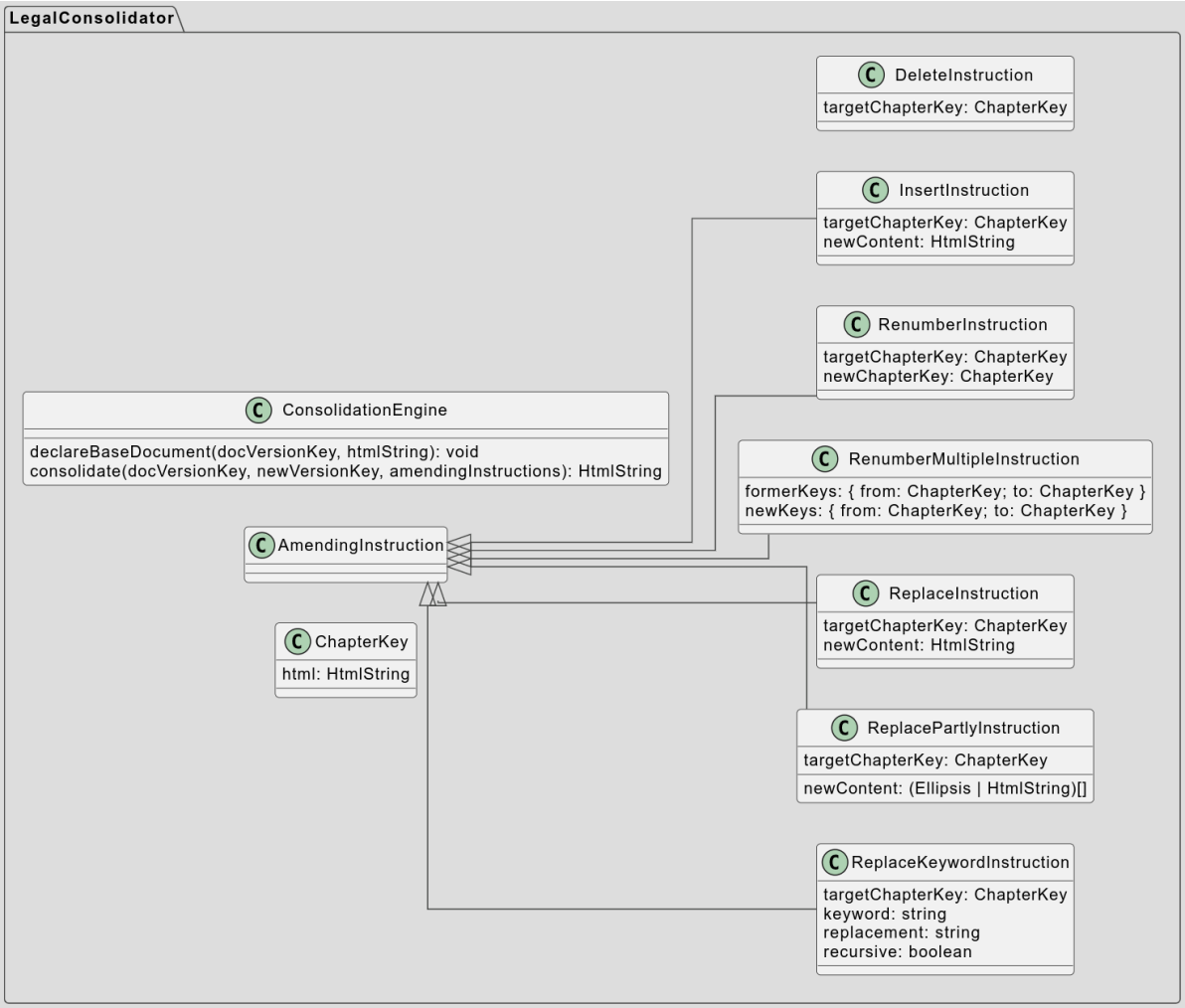


Figure 5.2.: UML class diagram of the public API of the legal-consolidator NPM package

5.1.3. AmendingInstruction

The AmendingInstruction type is a union type of the following different instruction types and their corresponding attributes:

DeleteInstruction

- *type: "DELETE"*
  In each DeleteInstruction, this field is set to the string literal "DELETE". This is done for type discrimination purposes and works the same across all other instruction types.
- *targetChapterKey: ChapterKey*
  Chapter key of the chapter to be deleted.

InsertInstruction

- *type: "INSERT"*
- *targetChapterKey: ChapterKey*
  Chapter key of the newly introduced chapter.
- *newContent: HtmlString*

RenumberInstruction

- *type: "RENUMBER"*
- *targetChapterKey: ChapterKey*
  Chapter key of the chapter to be renumbered.
- *newChapterKey: ChapterKey*
  New key that is to be assigned to the target chapter.

RenumberMultipleInstruction

- *type: "RENUMBER_MULTIPLE"*
- *formerKeys:*
  - *from: ChapterKey*
  - *to: ChapterKey*
- *newKeys:*
  - *from: ChapterKey*
  - *to: ChapterKey*

An example of how a RenumberMultipleInstruction could be encoded in JSON could look like the following:

Modificatory Provision:

Paragraphs 2.29. to 2.53., renumber as *2.30. to 2.54., respectively.*

JSON Encoding:

```
{
    "type": "RENUMBER_MULTIPLE",
    "formerKeys": {
        "from": {
            "value": "2.29"
        },
        "to": {
            "value": "2.53"
        }
    },
    "newKeys": {
        "from": {
            "value": "2.30"
        },
        "to": {
            "value": "2.54"
        }
    }
}
```

Figure 5.3.: Example of renumbering multiple chapters at once. Here, all chapters from 2.29. to 2.53. are incremented by one. [UNR160/0.1.0/proposed]

ReplaceInstruction This instruction replaces the content of a chapter completely with the provided new text content.

- *type: "REPLACE"*
- *targetChapterKey: ChapterKey*
- *newContent: HtmlString*

ReplacePartlyInstruction This instruction enables the use of ellipses in order to partially replace the original text. The consolidation engine will try to find the corresponding text from the base document, which is supposed to be inserted in the place of the ellipses, and will then translate this instruction into a simple ReplaceInstruction.

- *type: "REPLACE_PARTLY"*
- *targetChapterKey: ChapterKey*
- *newContent: ("..." | HtmlString)[]*
  Array of either text content or ellipsis literals ("...").

Modificatory Provision:

Paragraph 6.2.1.2., amend to read:

"6.2.1.2. If a device for indirect vision …… the total process of scanning, rendering and reset to its initial position together shall not take more than 200 milliseconds at room temperature of 22 °C ± 5 °C."

JSON Encoding:

```
{
    "type": "REPLACE_PARTLY",
    "newContent": [
        "6.2.1.2. If a device for indirect vision",
        "...",
        "the total process of scanning, rendering and reset to its initial
        ↪ position together shall not take more than 200 milliseconds at
        ↪ room temperature of 22 °C ± 5 °C."
    ],
    "targetChapterKey": {
        "value": "6.2.1.2."
    }
}
```

Figure 5.4.: Example of a partial replacement using ellipses. [UNR46/4.4.0/proposed]

ReplaceKeywordInstruction Oftentimes, modificatory provisions will instruct to replace all occurrences of a certain keyword with a replacement keyword. The implementation provides an additional field, which specifies whether this replacement operation is to be applied to all child chapters as well.

- *type: "REPLACE_KEYWORD"*
- *targetChapterKey: ChapterKey*
- *keyword: string.*
- *replacement: string*
- *recursive: boolean*

Modificatory Provision:

Through all the text of the Regulation, the acronym RESS, correct to read REESS

JSON Encoding:

```
{
    "type": "REPLACE_KEYWORD",
    "targetChapterKey": {
        "value": "ROOT"
    },
    "keyword": "RESS",
    "replacement": "REESS",
    "recursive": true
}
```

Modificatory Provision:

Annex 2, paragraph 2.1., replace the words "in the roof" by "on the ceiling".

JSON Encoding:

```
{
    "type": "REPLACE_KEYWORD",
    "targetChapterKey": {
        "value": "ROOT"
    },
    "keyword": "in the roof",
    "replacement": "on the ceiling",
    "recursive": false
}
```

Figure 5.5.: Examples recursive and non-recursive keyword replacement. [UNR46/4.4.0/proposed (top) UNR118/2.4.0/proposed]

5.1.4. ChapterKey

The chapter key models the requirements laid out in chapter 4.4.2. In accordance with this, the railroad diagram for the string representation of a ChapterKey is described in the following railroad diagram:

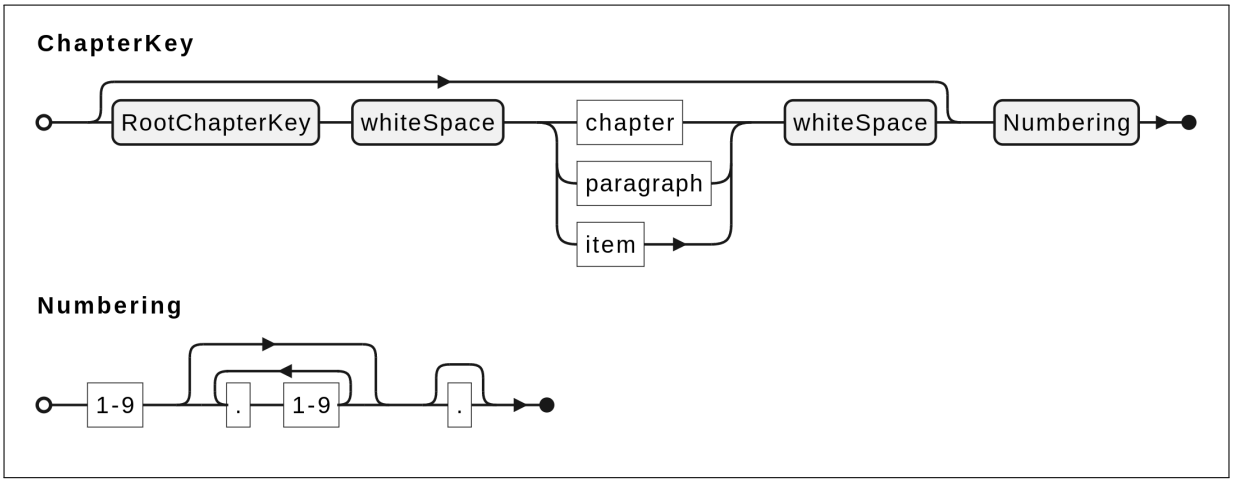


Figure 5.6.: Syntax definition for chapter keys required by the consolidation engine reference implementation.

Attributes

- *value: string*
  String representation of the chapter key.
- *getParentKey(): ChapterKey | undefined*
  Returns the parent key. If the current chapter is already a root chapter, its parent is undefined.
- *getYoungerSiblingKey(): ChapterKey | undefined*
  Returns the key of the current chapter's younger sibling. Returns undefined if the current chapter is its parent's first child.

Since numbering alone is often not enough to ensure document-level uniqueness, it often becomes necessary to specify the root chapter along with the numbering. Examples of valid string representations of chapter keys include the following:

- 1.
- 5.4.3.1.
- 5.4.3.1
- Annex 3 - Appendix 1 chapter 3.12.

5.1.5. declareBaseDocument()

This method provides a way of registering a base document version with the consolidation engine. The provided key can be later used in order to target this base document with an amending document.

Parameters:

- *documentVersionKey: DocumentKey*
  Specifies the key with which the document can be later targetted with an amending document.
- *baseDocument: HtmlString*
  A string containing the base document representation in HTML format as described in chapter 5.1

5.1.6. consolidate()

This method provides a way of registering a base document version with the consolidation engine. The provided key can be later used in order to target this base document with an amending document.

Parameters:

- *documentVersionKey: DocumentKey*
  The key corresponding with the base document upon which the amending instructions are to be applied.
- *instructions: AmendingInstruction[]*
  A list of amending instructions that are to be applied.

Output:

- *consolidatedDocument: HtmlString*
  Resulting consolidated document in HTML format, on which all instructions were applied.

5.2. User Interface - Amendment Editor

The main artifact result of this thesis, from a software perspective, is the consolidation engine. However, a way of interfacing with this consolidation engine is also developed. This is especially important since during the evaluation of the consolidation engine's performance, large amounts of amending provisions will need to be (largely manually) transformed into a suitable format, that the consolidation engine can consume. As such, we envision the sequence diagram for this system to look like the following:

Figure 5.7.: Certivity document representation.

5.2.1. Tauri

Tauri is an open-source, Rust-based framework that enables developers to write native desktop applications using web technologies such as HTML, CSS, and JavaScript. As of this writing, Tauri supports the use of any front-end JavaScript framework and is able to compile the resulting program into either installers or bundled programs. These are able to run on macOS (*.app, *.dmg), Windows (*.msi), and most Linux distributions (*.deb, *.AppImage).

Generally, Tauri works very similarly to Electron, an established framework, being used in production for programs like Spotify, Microsoft Teams, Discord, and Slack. These programs only have to be developed once and can be deployed as both web apps and native desktop clients. However, Tauri still offers a variety of benefits in comparison to Electron while still being mature enough to be used productively:

- **Bundle size**: Tauri bundles/installers are often only a fraction of their Electron counterparts by a factor of 15-30x.
- **Launch time**: Electron apps usually suffer from long startup times. While Tauri may not be as fast in comparison to an authentic native desktop app, it still outperforms Electron up to 2x.
- **Runtime performance**: Similar to the launch time, Electron and Tauri apps can be quite heavy on RAM usage since they basically require a browser to run in tandem with the program. Tauri however still manages to come in with 50% less RAM usage.

The main reason for choosing a native application framework like Tauri is the elevated access to the local file system which would be more difficult with a pure web app. In addition to this, the framework provides flexibility for the deployment process. Tauri ensures that there will be no need for hosting the web app long after the thesis has concluded. Rather, interested parties can simply download the pre-built program from a repository and run it directly. The ability to host the Amendment Editor on the web is not affected by this and can be done in

parallel.
[Lev][Asa][Con]

5.2.2. Input Methods

The current implementation of the Amendment Editor uses the Monaco editor library as the input interface for the user. This library is also utilized as the basis for the popular code editor Visual Studio Code.

The tool uses this editor for the user to input JSON data, which can be deserialized into AmendingInstruction objects, outlined in chapter 5.1.3. Under the hood, we perform JSON schema validation to provide code completion and error messages. The default code formatting of the Monaco editor library is used to format the code automatically.

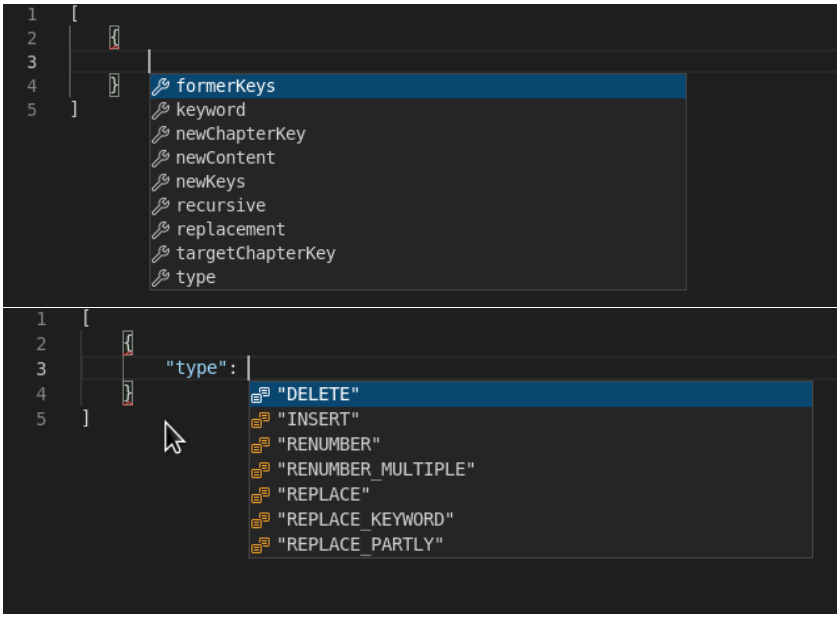


Figure 5.8.: Examples of code completion enabled by JSON schema.

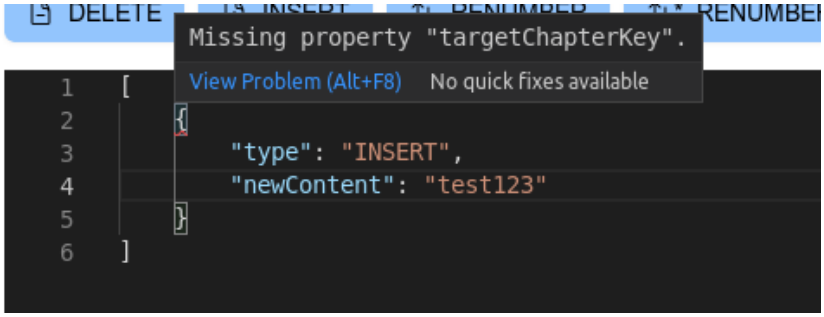


Figure 5.9.: Examples of error messages provided by JSON schema validation.

[Fou]

6. Evaluation

6.1. Experiment Design

This chapter describes the expressiveness of the encoding laid out in chapters 4.4 & 4.3 as well as the correctness of the consolidated documents created by the reference implementation of the consolidation engine (further detailed in chapter 5).

A UI was developed to facilitate and accelerate the ingestion of regulatory documents as well as preview the resulting consolidated document (chapter 5.2). This tool has been used during the experiment to speed up the manual conversion of unstructured PDF data into a machine-readable and -executable format in JSON and to automatically generate the consolidated documents in HTML.

The results of this automatically performed consolidation were then compared against the data set of consolidated documents, which were manually consolidated by regulatory experts at Certivity as the ground truth. Lastly, cases were categorized and analyzed, in which either the suggested formalization format was not expressive enough to encode the modificatory provision or in which the reference implementation of the consolidation engine has not produced the correct consolidated artifact even with the correct formalization.

Data Set Description For any regulation to be viable for this experiment, they need to fulfill the following requirements:

- Base document version available in HTML, with annotations for chapter borders and identifiers
- Consolidated documents available in HTML
- Amending documents for this regulation available in any format

Since the data set provided by Certivity only includes one amending document for each American regulation and usually multiple amending documents for each regulation in the UN data, 10 regulations from the UN data set and 25 regulations from the US data set were chosen. In total, the artifacts chosen for this experiment include:

	Regulations	Amending Documents	Instructions
UN	10	53	334
US	25	25	245

Table 6.1.: Breakdown of the amount of regulatory document artifacts considered in the experiment.

A full list of the regulations used in this experiment and their title is provided in addendum A.1. The titles can provide context about the more specific contents of each regulation.

6.2. Classification of Problematic Modificatory Provisions

In the following chapter, we discuss different modificatory provisions that were either impossible to be translated into a machine-executable format or were not able to be applied to the base document version correctly by the consolidation engine. Some examples of modificatory provisions that caused issues during the experiment were classified broadly into the following categories.

6.2.1. Formalization Format not Expressive Enough

The following describes modificatory provisions containing information that is impossible to translate into our machine-executable format due to them containing information that is not able to be modeled with our current format described in chapter 4.3.

Numbering Removal In this example, the amending document instructs to remove the numbering of a certain chapter and to amend its content.

Paragraph 6.4.1., delete the numbering and amend to read (including the addition of two tables):

"The subject vehicle …

Tests shall be conducted with a vehicle travelling at speeds shown in the tables below for respectively M_1 and N_1 Categories. If this is deemed justified, the technical service may test any other speeds listed in the tables in paragraph 5.2.1.4. and within the prescribed speed range as defined in paragraph 5.2.1.3.

Subject vehicle test speed for M_1 category in stationary target scenario

Maximum mass	*Mass in running order*	*Tolerance*
20	20	+2/-0
40	42	+0/-2
60	60	+0/-2

Subject vehicle test speed for N_1 category in stationary target scenario

Maximum mass		*Mass in running order*		*Tolerance*
α >1.3	*α ≤1.3*	*α >1.3*	*α ≤1.3*	
20	20	20	20	+2/-0
38	30	42	35	+0/-2
60	60	60	60	+0/-2

The functional part …"

Figure 6.1.: Example of a numbering removal instruction. [UNR152 0.3.0]

The intended effect of this instruction is to attach the contents of this chapter to the end of the previous chapter. This could either be an older sibling of the target chapter or its parent, depending on whether it is the first child of its parent.

Sentence-level Targeting In this instance, the amending instruction targets only the second sentence of a particular chapter, leaving the remaining parts of the text untouched. Sentence-level targeting is not supported in the current formalization format.

■ 3. Section 571.224 is amended by revising the second sentence in S3 and the definition of ‘‘Rear extremity’’ in S4 to read as follows:

Figure 6.2.: Example of sentence-level targeting. [§571.224 0.5.0]

6.2.2. Insufficient/Erroneous Reference Implementation

This error archetype is for unexpected behavior caused not by the translation from natural language into our formalized machine-executable format, but rather during its application to the document to be amended.

Keyword Inflection Replacement In this example, the modificatory provision instructs to replace all occurrences of "Rechargeable Energy Storage System (REESS)" with "Rechargeable

Electrical Energy Storage System (REESS)", essentially adding in the extra word "Electrical". However, since the current implementation of the consolidation engine performs the keyword replacement based on string literal matches it struggles with replacing inflections of the keyword. In this case, every occurrence of the keyword to be replaced in the document does not match the given keyword literal.

Rather,

- the first occurrence is a pluralization of the keyword
- the second occurrence has a different capitalization
- the third occurrence is missing the "(RESS)", and
- the fourth occurrence has different capitalization yet again.

Modificatory Provision:

> *Through all text of the Regulation (including annexes)*, *Rechargeable Energy Storage System (REESS)*, amend to read: Rechargeable Electrical Energy Storage System (REESS).

Target Text:

> 2.2.2.2. The locations of the Rechargeable Energy Storage Systems (REESS), in so far as they have a negative effect on the result of the impact test prescribed in this Regulation;
>
> 2.20. "*Rechargeable energy storage system (REESS)*" means rechargeable energy storage system which provides electrical energy for propulsion;
>
> 2.29. "*Coupling system for charging the rechargeable energy storage system (REESS)*" means the electrical circuit used for charging the REESS from an external electrical power supply including the vehicle inlet;

Intended Result:

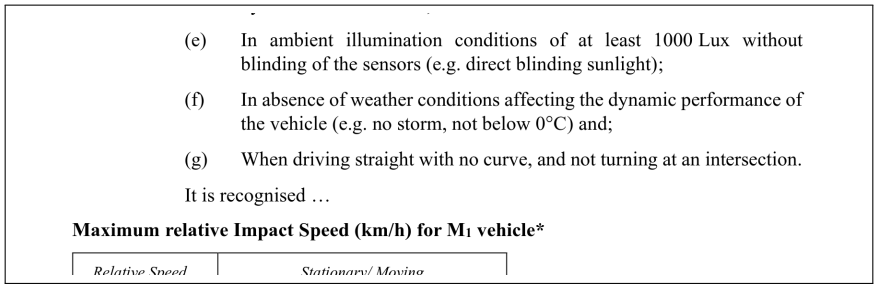
> 2.2.2.2. The locations of the Rechargeable Electrical Energy Storage Systems (REESS), in so far as they have a negative effect on the result of the impact test prescribed in this Regulation;
>
> 2.20. "*Rechargeable electrical energy storage system (REESS)*" means rechargeable electrical energy storage system which provides electrical energy for propulsion;
>
> 2.29. "*Coupling system for charging the rechargeable electrical energy storage system (REESS)*" means the electrical circuit used for charging the REESS from an external electrical power supply including the vehicle inlet;

Figure 6.3.: Example of keyword inflection causing problems for REPLACE_KEYWORD instructions. [UNR12 4.4.0]

Capitalization issues are easily solvable via normalization of the capitalization to all lowercase. Other types of inflections could potentially be dealt with by stemming/lemmatizing the original text to be amended in order to perform better matching with the keyword. The issue of how to adjust the replacement to make it grammatically match the original text however remains quite a difficult topic to resolve.

Failed Ellipsis Replacement In these cases, the consolidation engine did not manage to replace ellipses from the amending instruction with the correct text from the target document.

Modificatory Provision:



(e) In ambient illumination conditions of at least 1000 Lux without blinding of the sensors (e.g. direct blinding sunlight);

(f) In absence of weather conditions affecting the dynamic performance of the vehicle (e.g. no storm, not below 0°C) and;

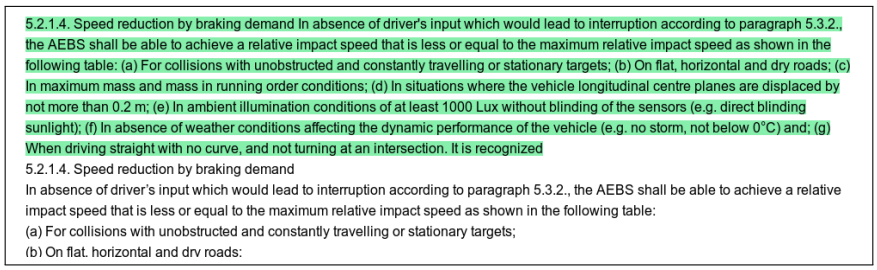
(g) When driving straight with no curve, and not turning at an intersection.

It is recognised …

Maximum relative Impact Speed (km/h) for M_1 vehicle*

Relative Speed	*Stationary/ Moving*

Reference Implementation Output:



5.2.1.4. Speed reduction by braking demand In absence of driver's input which would lead to interruption according to paragraph 5.3.2., the AEBS shall be able to achieve a relative impact speed that is less or equal to the maximum relative impact speed as shown in the following table: (a) For collisions with unobstructed and constantly travelling or stationary targets; (b) On flat, horizontal and dry roads; (c) In maximum mass and mass in running order conditions; (d) In situations where the vehicle longitudinal centre planes are displaced by not more than 0.2 m; (e) In ambient illumination conditions of at least 1000 Lux without blinding of the sensors (e.g. direct blinding sunlight); (f) In absence of weather conditions affecting the dynamic performance of the vehicle (e.g. no storm, not below 0°C) and; (g) When driving straight with no curve, and not turning at an intersection. It is recognized

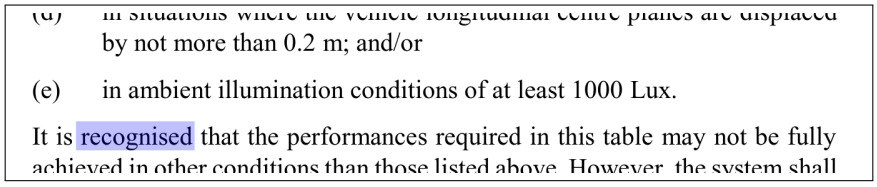
5.2.1.4. Speed reduction by braking demand

In absence of driver's input which would lead to interruption according to paragraph 5.3.2., the AEBS shall be able to achieve a relative impact speed that is less or equal to the maximum relative impact speed as shown in the following table:

(a) For collisions with unobstructed and constantly travelling or stationary targets;

(b) On flat, horizontal and dry roads:

Target Text:

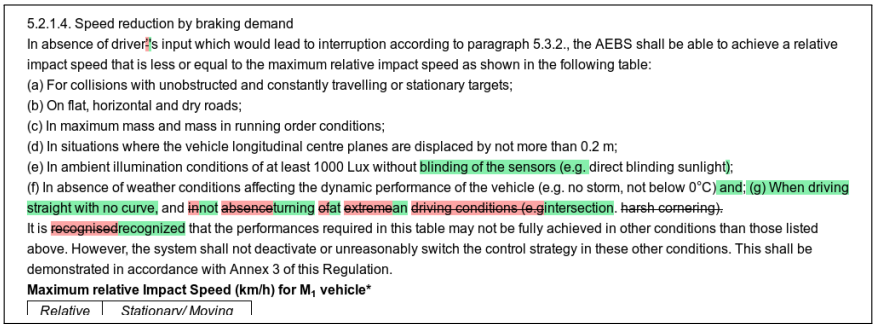


(d) in situations where the vehicle longitudinal centre planes are displaced by not more than 0.2 m; and/or

(e) in ambient illumination conditions of at least 1000 Lux.

It is recognised that the performances required in this table may not be fully achieved in other conditions than those listed above. However, the system shall

Intended Result:



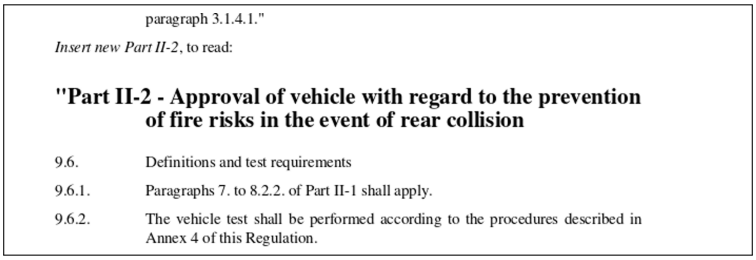
5.2.1.4. Speed reduction by braking demand
In absence of driver's input which would lead to interruption according to paragraph 5.3.2., the AEBS shall be able to achieve a relative impact speed that is less or equal to the maximum relative impact speed as shown in the following table:
(a) For collisions with unobstructed and constantly travelling or stationary targets;
(b) On flat, horizontal and dry roads;
(c) In maximum mass and mass in running order conditions;
(d) In situations where the vehicle longitudinal centre planes are displaced by not more than 0.2 m;
(e) In ambient illumination conditions of at least 1000 Lux without blinding of the sensors (e.g. direct blinding sunlight);
(f) In absence of weather conditions affecting the dynamic performance of the vehicle (e.g. no storm, not below 0°C) and; (g) When driving straight with no curve, and inot absenceturning ofat extremean driving conditions (e.gintersection. harsh cornering).
It is recognisedrecognized that the performances required in this table may not be fully achieved in other conditions than those listed above. However, the system shall not deactivate or unreasonably switch the control strategy in these other conditions. This shall be demonstrated in accordance with Annex 3 of this Regulation.
Maximum relative Impact Speed (km/h) for M_1 vehicle*
Relative | Stationary/ Moving

Figure 6.4.: Example of failed ellipsis replacement due to inconsistent spelling. [UNR12 4.4.0]

In this particular example, "It is recognized..." was not enough context for the consolidation engine to find the correct corresponding text due to the different spelling of "recognized" as opposed to "recognised". This could arguably be counted as a mistake by the amending author as he chose a different spelling than the original document. This change is very likely to be unintended as it does not have any semantic effect on the sentence. However, a better consolidation engine implementation could potentially be able to match words based on their semantics rather than their spelling alone in order to better handle cases such as this.

Insertion of Chapter with Non-conforming Chapter Key Some identifiers do not follow the requirements of unique chapter identifiers defined in chapters 4.4.2 and 5.1.4.

The implementation of chapter keys laid out in chapter 5.1.4 does not support these types of keys properly and the consolidation thus does not know where to insert this chapter.



paragraph 3.1.4.1."

Insert new Part II-2, to read:

"Part II-2 - Approval of vehicle with regard to the prevention of fire risks in the event of rear collision

9.6. Definitions and test requirements

9.6.1. Paragraphs 7. to 8.2.2. of Part II-1 shall apply.

9.6.2. The vehicle test shall be performed according to the procedures described in Annex 4 of this Regulation.

Figure 6.5.: Example of an insertion of a chapter with a non-conforming chapter key. [UNR34 3.0.0]

This description only pertains to insertions of chapters with non-conforming identifiers. Issues with targeting existing chapters with non-conforming identifiers for amendments are described in chapter 6.2.3.

6.2.3. Data Set Problems

Some errors were caused by the base document versions not having sufficient metadata annotation to perform some of the actions stated in the amending documents.

Lacking Sub-chapter Entity Annotation Most of the issues found in the US data set had to do with insufficient data labeling/annotation of the base document versions. A lot of chapters in the US data set contain sub-sections that do not follow the usual numbering scheme described in chapter 5.1.4.

Modificatory Provision:

■ 7. Amend § 571.206 by revising paragraph S2, the definitions of ''Side Front Door'' and ''Side Rear Door'' in paragraph S3, and paragraph S5.1.1.4(b)(1)(ii)(C) to read as follows:

§ 571.206 Standard No. 206; Door locks and door retention components.

* * * * *

S2. *Application.* This standard

Target Text:

S5.1.1.4 *Inertial Force Application.* The test procedures for S4.1.1.4 and S4.2.1.3 are as follows:

(a) *Calculation.* The calculation is performed in accordance with paragraph 6 of SAE Recommended Practice J839 (1991) (incorporated by reference, see § 571.5).

(b) *Dynamic Test.* The dynamic inertial force application is tested according to the setup specified in paragraph (1) or (2) of this section.

(1) *Test Setup and Directions for Full Vehicle Test.*

(i) Test Setup.

(A) Rigidly secure the full vehicle to an acceleration device that, when accelerated together will assure that all

Figure 6.6.: Example of a modificatory provision targeting a deeply nested sub-chapter entity: S5.1.1.4(b)(1)(ii)(C) §571.206

In the above example, chapter S5.1.1.4 is segmented into smaller sections resulting in the following hierarchy:

```
- S5.1.1.4
    - (a) Calculation
    - (b) Dynamic Test
        - (1) Test Setup and Directions for Full Vehicle Setup
            - (i) Test Setup
                - (A) Rigidly...
                - (B) Install the...
                - ...
            - (ii) Test Directions
            - ...
        - (2) Test Setup and Directions for Door Test
        - ...
```

Figure 6.7.: Example of sub-chapter nesting in the US data set.

The base document versions in the data set did not have these sub-chapter numberings metadata annotated. As a result, the whole chapter S5.1.1.4 is treated as one chapter with no children. The information about the inner hierarchy of S5.1.1.4 is lost and the modificatory provision could not be properly evaluated. Several instructions were not able to be converted correctly into a machine-executable format for this reason.

This type of error is only pertinent to the amendment of existing sub-chapter entities within a document. The issues that arise when inserting new chapters with non-conforming identifiers are described in chapter 6.2.2.

6.2.4. Human Error by the Regulatory Body

Since the modificatory provisions contained in the amending documents are written by humans, they contain several types of human errors which make their translation into a machine-executable format challenging. In the context of this experiment and since these modificatory provisions were converted manually, these types of errors can usually be handled fairly easily by humans. However, when working in a fully automated system, these conversions from amending documents in natural language into a machine-executable format would be significantly harder to perform.

Misleading Instructions In this example, the amending document instructs to renumber 21.11. to 21.16. as 21.10. to 21.15., i.e. decrementing each of the targetted chapters by one. However, the modificatory provisions provided in the example are in fact the only modificatory provisions in the whole amending document, making the operations impossible. This is due to the fact that no chapter from 21.1. to 21.10. have been deleted or renumbered, thus not leaving enough room for 21.11. to be decremented or else sacrificing internal document correctness as there would now be two chapters with the same identifier.

Modificatory Provision:

Paragraphs 21.11. to 21.16. (former), replace by:

Paragraphs 21.11. to 21.16. (former), re-number as paragraphs 21.10. to 21.15. and in *paragraph 21.16. (former),* correct the reference to paragraph 21.15. to read paragraph 21.14.

Target Text:

21.10. Notwithstanding the provisions of paragraphs 21.2., 21.4. and 21.5. above, for the purpose of replacement parts Contracting Parties applying this Regulation shall continue to grant approvals according 02 series of amendments to this Regulation, to devices for indirect vision for use on vehicle types which have been approved before the date mentioned in paragraph 21.2. above pursuant to the 02 series of amendments of Regulation No. 46, and, where applicable, subsequent extensions to these approvals.

21.11. As from the official date of entry into force of the 04 series of amendments to this Regulation, no Contracting Party applying this Regulation shall refuse an application for approval under this Regulation as amended by the 04 series of amendments.

21.12. As from 30 June 2014, Contracting Parties applying this Regulation shall grant approvals to a type of device for indirect vision only if the type of device meets the requirements of this Regulation as amended by the 04 series of amendments.

21.13. As from 30 June 2014, Contracting Parties applying this Regulation shall grant approvals to a type of vehicle with regard to the installation of devices for indirect vision only if the type of vehicle meets the requirements of this Regulation as amended by the 04 series of amendments.

21.14. As from 30 June 2015, Contracting Parties applying this Regulation shall not be obliged to accept approvals of a type of vehicle or type of device for indirect vision which have not been granted in accordance with the 04 series of amendments to this Regulation.

21.15. Notwithstanding paragraph 21.15. above, type approvals granted to the preceding series of amendments to the Regulation, which are not affected by the 04 series of amendments, shall remain valid and Contracting Parties applying this Regulation shall continue to accept them.

21.16. Contracting Parties applying this Regulation shall not refuse to grant extensions of type approvals for existing types of vehicles or devices, which are not affected by the 04 series of amendments, granted according to the 02 or 03 series of amendments to this Regulation.

21.17. Notwithstanding the provisions of paragraphs 21.2., 21.4., 21.5., 21.13. and 21.15. above, for the purpose of replacement parts, Contracting Parties applying this Regulation shall continue to grant approvals according to the 01 series of amendments to this Regulation, to devices for indirect vision of classes I to V for use on vehicle types which have been approved before 26 January 2006 pursuant to the 01 series of amendments of Regulation No. 46 and, where applicable, subsequent extensions to these approvals.

Intended Result:

21.10. Notwithstanding the provisions of paragraphs 21.2., 21.4. and 21.5. above, for the purpose of replacement parts Contracting Parties applying this Regulation shall continue to grant approvals according 02 series of amendments to this Regulation, to devices for indirect vision for use on vehicle types which have been approved before the date mentioned in paragraph 21.2. above pursuant to the 02 series of amendments of Regulation No. 46, and, where applicable, subsequent extensions to these approvals.	●	21.10. Notwithstanding the provisions of paragraphs 21.2., 21.4. and 21.5. above, for the purpose of replacement parts Contracting Parties applying this Regulation shall continue to grant approvals according 02 series of amendments to this Regulation, to devices for indirect vision for use on vehicle types which have been approved before the date mentioned in paragraph 21.2. above pursuant to the 02 series of amendments of Regulation No. 46, and, where applicable, subsequent extensions to these approvals.
21.11. As from the official date of entry into force of the 04 series of amendments to this Regulation, no Contracting Party applying this Regulation shall refuse an application for approval under this Regulation as amended by the 04 series of amendments.	●	21.11. As from the official date of entry into force of the 04 series of amendments to this Regulation, no Contracting Party applying this Regulation shall refuse an application for approval under this Regulation as amended by the 04 series of amendments.
21.12. As from 30 June 2014, Contracting Parties applying this Regulation shall grant approvals to a type of device for indirect vision only if the type of device meets the requirements of this Regulation as amended by the 04 series of amendments.	●	21.12. As from 30 June 2014, Contracting Parties applying this Regulation shall grant approvals to a type of device for indirect vision only if the type of device meets the requirements of this Regulation as amended by the 04 series of amendments.
21.13. As from 30 June 2014, Contracting Parties applying this Regulation shall grant approvals to a type of vehicle with regard to the installation of devices for indirect vision only if the type of vehicle meets the requirements of this Regulation as amended by the 04 series of amendments.	●	21.13. As from 30 June 2014, Contracting Parties applying this Regulation shall grant approvals to a type of vehicle with regard to the installation of devices for indirect vision only if the type of vehicle meets the requirements of this Regulation as amended by the 04 series of amendments.
21.14. As from 30 June 2015, Contracting Parties applying this Regulation shall not be obliged to accept approvals of a type of vehicle or type of device for indirect vision which have not been granted in accordance with the 04 series of amendments to this Regulation.	●	21.14. As from 30 June 2015, Contracting Parties applying this Regulation shall not be obliged to accept approvals of a type of vehicle or type of device for indirect vision which have not been granted in accordance with the 04 series of amendments to this Regulation.
21.15. Notwithstanding paragraph 21.15. above, type approvals granted to the preceding series of amendments to the Regulation, which are not affected by the 04 series of amendments, shall remain valid and Contracting Parties applying this Regulation shall continue to accept them.	●	21.15. Notwithstanding paragraph 21.1514. above, type approvals granted to the preceding series of amendments to the Regulation, which are not affected by the 04 series of amendments, shall remain valid and Contracting Parties applying this Regulation shall continue to accept them.
21.16. Contracting Parties applying this Regulation shall not refuse to grant extensions of type approvals for existing types of vehicles or devices, which are not affected by the 04 series of amendments, granted according to the 02 or 03 series of amendments to this Regulation.	●	21.16. Contracting Parties applying this Regulation shall not refuse to grant extensions of type approvals for existing types of vehicles or devices, which are not affected by the 04 series of amendments, granted according to the 02 or 03 series of amendments to this Regulation.
21.17. Notwithstanding the provisions of paragraphs 21.2., 21.4., 21.5., 21.13. and 21.15. above, for the purpose of replacement parts, Contracting Parties applying this Regulation shall continue to grant approvals according to the 01 series of amendments to this Regulation, to devices for indirect vision of classes I to V for use on vehicle types which have been approved before 26 January 2006 pursuant to the 01 series of amendments of Regulation No. 46 and, where applicable, subsequent extensions to these approvals.	●	21.17. Notwithstanding the provisions of paragraphs 21.2., 21.4., 21.5., 21.13. and 21.15. above, for the purpose of replacement parts, Contracting Parties applying this Regulation shall continue to grant approvals according to the 01 series of amendments to this Regulation, to devices for indirect vision of classes I to V for use on vehicle types which have been approved before 26 January 2006 pursuant to the 01 series of amendments of Regulation No. 46 and, where applicable, subsequent extensions to these approvals.

Figure 6.8.: Example of a misleading modificatory provision. [UNR46 4.1.1]

Regulatory experts at Certivity have interpreted this provision to simply state that only chapter 21.15. is to be amended.

Inaccurate Ellipsis Usage When using ellipses to partially replace a certain chapter as discussed in chapter 4.3.4.2, it is important for the author to provide enough context for the consolidation engine to match the correct parts of the text from the original document, which is to take the place of the ellipsis.

Modificatory Provision:

Paragraph 5.5.1., amend to read:

"5.5.1. Protection against electrical shock

…

If the test is performed under the condition that part(s) of the high voltage system are not energized, the protection against electrical shock shall be proved by either paragraph 5.5.1.3. or paragraph 5.5.1.4. below for the relevant part(s).

For the coupling system for charging the REESS, which is not energized during driving conditions, at least one of the four criteria specified in paragraphs 5.5.1.1. to 5.5.1.4. below shall be met."

Target Text:

protection IPXXB.

In the case that the test is performed under the condition that part(s) of the high voltage system are not energized, the protection against electrical shock shall be proved by either paragraph 5.5.1.3. or paragraph 5.5.1.4. for the relevant part(s).

Intended Result:

protection ~~degree~~ IPXXB.

~~In~~ If the ~~case that the~~ test is performed under the condition that part(s) of the high voltage system are not energized, the protection against electrical shock shall be proved by either paragraph 5.5.1.3. or paragraph 5.5.1.4. for the relevant part(s).

For the coupling system for charging the REESS, which is not energized during driving conditions, at least one of the four criteria specified in paragraphs 5.5.1.1. to 5.5.1.4. shall be met.

Figure 6.9.: Example of a misused ellipsis [UNR12 4.3.0]

In the above example and without a deeper semantic understanding of the text, it is close to impossible for the consolidation engine to recognize that "If the text is performed" is supposed to match with "In the case that the test is performed" from the original document text. It seems as though this was an unintended change in the first place and this example is therefore classified as a mistake from the amending document author.

Another example of inaccurate ellipsis usage can be shown in the following:

Modificatory Provision:

Paragraph 5.2.1.4., amend to read (addition of "and" at the end of item (f) in the list of conditions):

"5.2.1.4. Speed reduction by braking demand

In absence of driver's input …

…

(f) In absence of weather conditions affecting the dynamic performance of the vehicle (e.g. no storm, not below 0°C); and

(g) When driving straight with no curve, and not turning at an intersection.

It is recognised …"

Intended Result:

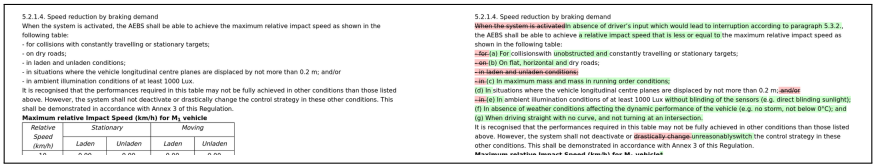


Figure 6.10.: Example of a misused ellipsis [UNR152 0.4.0]

Forgotten Ellipsis In this modificatory provision, the amending document author simply forgot to include the ellipsis at the end, as the chapter is supposed to continue after "Testing of articulated vehicles".

Modificatory Provision:

Paragraph 5.5., amend to read:

"5.5. Testing of articulated vehicles"

Intended Result:

Figure 6.11.: Example of a missing ellipsis [UNR66 2.0.0]

This case should not be hard to handle for humans, since it is very unlikely with this wording that the amending author intended the consolidator to delete the rest of the chapter. An automated system, however, would have trouble making this decision.

As the counterpart to this, sometimes unnecessary ellipses are included. Although, if placed at the start or end of the text, these do not cause any problems, since the consolidation engine still handles this case correctly. Example:

Paragraph 5.2.1.1., amend to read:

"5.2.1.1. An immobilizer shall be designed so as to prevent the operation of the vehicle under its own motive power by at least one of the following means:

…"

Figure 6.12.: Example of an unnecessary ellipsis since the chapter ends after "following means:" [UNR162 0.2.0]

Wrong Chapter Numbering Sometimes, authors of the amending document make a mistake and apply the wrong numbering as showcased in the example below. For a human, it is clear that the third chapter to be inserted is actually labeled incorrectly and should be numbered 2.16. This is because the text clearly states to insert "chapters 2.14. to 2.16." and the fact that 2.14. and 2.15. should naturally be followed by 2.16. However, an automated system would have a really hard time spotting that this is an error in the first place in addition to having to make a decision of which value should be the correct one.

Modificatory Provision:

Insert new paragraphs 2.14. to 2.16., to read:

"2.14. "Primary user" is a user who is able to authorize digital keys. There can be more than one primary users.

2.15. "Digital key" means a key designed to be transferred to multiple devices by the primary user(s) through dedicated processes.

2.12. "Close proximity" means a distance of less than 6 m."

Figure 6.13.: Example of wrong numbering in the amending document. [UNR162 0.1.0]

Inconsistent Renumbering after Deletion Usually, in the UN data set, every renumbering instruction is explicit. This is especially important for instructions that affect not only the target chapter to be amended but also the numbering of chapters surrounding it.

These instances occur for example when inserting a new chapter between two existing sibling chapters. Every subsequent chapter after the newly inserted chapter would have to have their numbering incremented by one to account for the new chapter. Correspondingly, when deleting a chapter that has younger siblings, every younger sibling of the chapter to be deleted would have to have their numbering decremented by one.

Modificatory Provision:

Paragraphs 2.24. to 2.34.2., shall be deleted.

Target Text:

2.34. "*Brake Assist System (BAS)*" means a function of the braking system that deduces an emergency braking event from a characteristic of the driver's brake demand and, under such conditions:
(a) Assists the driver to deliver the maximum achievable braking rate; or
(b) Is sufficient to cause full cycling of the Anti-lock Braking System.

2.34.1. "*Category A Brake Assist System*" means a system which detects an emergency braking condition based primarily[3] on the brake pedal force applied by the driver;

[3]As declared by the vehicle manufacturer.

2.34.2. "*Category B Brake Assist System*" means a system which detects an emergency braking condition based primarily[3] on the brake pedal speed applied by the driver;

[3]As declared by the vehicle manufacturer.

2.35. "*Identification code*" identifies the brake discs or brake drums covered by the braking system approval according to this regulation. It contains at least the manufacturer's trade name or trademark and an identification number.

Intended Result:

~~As declared by the vehicle manufacturer.~~

~~2.34.2. "*Category B Brake Assist System*" means a system which detects an emergency braking condition based primarily[3] on the brake pedal speed applied by the driver;~~

~~[3]As declared by the vehicle manufacturer.~~

2.~~35~~24. "*Identification code*" identifies the brake discs or brake drums covered by the braking system approval according to this regulation. It contains at least the manufacturer's trade name or trademark and an identification number.

Figure 6.14.: Example of a misused ellipsis [UNR12 4.3.0]

In the above example, the amending author forgot to renumber 2.35. to 2.24. after deleting chapters 2.24. to 2.34. This would cause issues with future amendments since the numbering is no longer consistent.

6.2.4.1. Out-of-scope issues

Certain types of modificatory provisions were intentionally left out from the reference implementation for the sake of brevity. Although technically not considered faults in the system, the following showcases some interesting examples and suggestions for future research.

Targetting non-chapter entities As discussed in chapter 2.1.10, non-chapter entities are those which do not have a chapter number with which they are targetted. Operations that target non-chapter entities and are not supported by the consolidation engine reference implementation include:

Adding / Modifying Images In the current system, the only way for a user to declare the contents of any given chapter that is to be newly introduced is via HTML. The amendment

editor tool currently supports no way of introducing new chapters that contain images via INSERT or REPLACE instruction.

In order to facilitate this feature, one could either upload the image to a certain URL, then link to that URL in the HTML or encode the image in Base64 encoding and embed that data directly in the HTML.

Adding / Modifying Tables The base document versions provided by Certivity do not have labels for certain table sections that might be the target of modificatory provisions. Any modificatory provision that targets tables or sections of tables therefore could not be translated into our formalization format.

Adding / Modifying Footnotes The current implementation of the consolidation engine treats footnotes as part of the chapter in which they are referenced. Targeting only the footnote is therefore not currently supported.

6.3. Results Overview

This chapter will provide an overview of the evaluation results of both the consolidation engine and the formalization format for modificatory provisions.

	Total Instructions	**Formalization Succeeded**	**Automatic Application Correct**
UN	334	326 (97.6% of total)	317 (97.2% of formalized)
US	245	165 (67.3% of total)	165 (100% of formalized)

Table 6.2.: Breakdown of the success rate of the two experimentation steps categorized by region.

6.3.1. UN data set

Generally, modificatory provisions in the UN data set posed little issues both on translation into the formalization format proposed in this thesis and when automatically applying the formalized modificatory provisions to the target document. In total, **317 out of 334 (94.9%)** amending instructions showed no issues both on translation and automatic consolidation.

Using the classification defined in chapter 6.2, the remaining 17 amending instructions can be closer described in the following:

Formalization Format not Expressive Enough

- 1 case of an unsupported "insert non-numeric identifier" instruction
- 1 case of an unsupported "remove numbering" instruction

Insufficient/Erroneous Reference Implementation

- 1 case of inflected keyword replacement
- 4 cases of failed ellipsis replacement
  Here, the consolidation engine reference implementation failed to perform the correct ellipsis replacement.

Human Error by the Regulatory Body

- 1 case of inaccurate ellipsis
- 1 case of forgotten renumbering
- 1 case of forgotten ellipsis

Out-of-scope issues

- 5 picture operations (both inserting new and amending existing figures)
- 1 instruction to change the title of a figure
- 1 footnote operation

More detailed descriptions of these issues with examples from the original texts will be presented in chapter 6.2.

6.3.2. US data set

Similarly to the UN data set, basic modificatory provisions containing only simple INSERT, AMEND, and DELETE instructions were handled very reliably by the consolidation engine. The translation of these amending instructions to the formalization format also did not pose any significant challenge.

In contrast to the UN data set, it can be observed that the instruction formulations used in modificatory provisions from the US data set are more consistent, using only a small set of terms to describe the changes.

Overall, **146 out of 245 or 60%** of the US data did not present any issues when translating to the formalization format and when being automatically applied to the target document. When disregarding issues with sub-chapter targeting, this percentage grows to **88.6% or**

146 out of 165. The 98 amending instructions that caused issues can be broken down as follows:

Formalization Format not Expressive Enough

- 1 instruction to amend only the second sentence of a chapter

Data Set Problems

- 79 amending instructions targetting sub-chapter sections

Out-of-scope issues

- 12 instructions targeting definition items
- 4 instructions targeting pictures
- 3 instructions targeting tables

More detailed descriptions of these issues with examples from the original texts will be presented in chapter 6.2.

6.4. Discussion

Generally, performing basic replacements, insertions and deletions of numbered chapters in regulatory documents poses little challenge regarding automation. However, the numerous edge cases, implicit information, and mistakes made by the author of the modificatory provision make the automation exceedingly difficult, even with 100% correct information extraction of the amending document.

In the following, we discuss how the results of this thesis answer the research questions laid out in chapter 1.2.

6.4.1. RQ1: What is the minimum set of consolidation engine operations needed to model all modificatory provisions in the data set?

In theory, all modificatory provisions can be broken down into two very basic operations: insertions and deletions. Any other more complicated operation can be modeled with these two. A renumber operation for example can be reformulated as a delete operation with subsequent insertion of the same chapter with different numbering.

However, in the data set, more complex operations appear often enough to warrant closer examination and the formalization format has benefitted in readability and brevity from

supporting these special operations. Overall, the amending instructions identified and their potential target entities are listed in table **??**

Operations	**Potential Targets**					
	Chapters	Tables	Table Sections	Pictures	Footnotes	Definition Items
INSERT	✓	✓	✓	✓	✓	✓
DELETE	✓	✓	✓	✓	✓	✓
REPLACE	✓	✓	✓	✓	✓	✓
REPLACE_PARTLY	✓	✗	✗	✗	✓	✓
REPLACE_KEYWOR	✓	✓	✓	✓	✓	✓
RENUMBER	✓	✗	✗	✗	✗	✗
RENUMBER_MULTIPLE	✗	✗	✗	✗	✗	✗
REMOVE_NUMBERING	✗	✗	✗	✗	✗	✗

Table 6.3.: Overview of amending instruction occurrences in the data set (binary) together with their respective potential targets.

Note that this table exclusively documents occurrences in the data set. Renumbering a table (i.e. changing its identifier) for example could be possible in regulatory documents from another regulatory body. However, this did not occur in the documents examined during this thesis.

Closer examination of the different operations and their data requirements are documented in chapter 4.3.

6.4.2. RQ2: What is the minimum set of metadata fields needed in the representation formats of base documents and modificatory provisions in order to perform automatic consolidation?

As outlined more closely in chapter 4.4, the representation format is required to support a hierarchical tree structure to model the parent-child relationships that chapters have with each other correctly.

In addition to this and to support proper targeting, document entities that are potential targets of modificatory provisions need to be annotated in terms of their position in the text as well as a fitting identifier. In the data set, these document entities were targeted by modificatory provisions:

- Chapters
- Tables
- Table Sections
- Pictures
- Footnotes
- Definition Items

6.4.3. RQ3: How accurate are the automatic change applications performed by the consolidation engine reference implementation?

Overall, the results of the automatic change application are promising. Provided that the reference implementation consolidation engine is supplied with correct conversions of the modificatory provision in the formalization format, there were only 8 cases, in which the consolidation engine failed to provide the correctly consolidated output document. 3 of these cases can be traced back to human error by the regulatory bodies. The other 5 cases are discussed in more detail in chapter 6.2.2.

7. Conclusion

In conclusion, this master's thesis presents a novel approach to legal consolidation, specifically targeting documents released by the UNECE and the US federal government. A human-readable and -writable formalization format for modificatory provisions was developed as well as a consolidation engine reference implementation capable of applying a subset of these provisions automatically to a target document. The evaluation of both the formalization format and the consolidation engine demonstrated the potential of our approach in automating basic replacements, insertions, and deletions in regulatory documents.

However, the research also identified several challenges, often caused by implicit information, unusual edge cases, and mistakes made by regulatory bodies, currently preventing the full automation of the legal consolidation process. Despite these limitations, this thesis contributes to the ongoing efforts of automating legal consolidation by documenting difficult edge cases and highlighting promising areas for future research. By addressing the identified challenges and further refining the formalization format and consolidation engine, it is possible to significantly reduce the time and effort spent on legal consolidation tasks while minimizing errors in the process.

In light of these findings, future work should focus on improving the semantic information extraction and annotation, as well as incorporating this thesis' findings on edge cases and potential issues into the development of a more reliable and accurate solution. Furthermore, extending the scope of research beyond the Italian context and exploring the applicability of this approach to other languages and legal systems will be beneficial in contributing to the global advancement of fully automated legal consolidation.

A. General Addenda

A.1. List of Regulations Examined during Evaluation

- CFR Title 49 §571.105: Hydraulic and electric brake system
- CFR Title 49 §571.121: Air brake systems
- CFR Title 49 §571.122: Motorcycle brake systems
- CFR Title 49 §571.126: Electronic stability control systems for light vehicles
- CFR Title 49 §571.135: Light vehicle brake systems
- CFR Title 49 §571.136: Electronic stability control systems for heavy vehicles
- CFR Title 49 §571.139: New pneumatic radial tires for light vehicles
- CFR Title 49 §571.141: Minimum Sound Requirements for Hybrid and Electric Vehicles
- CFR Title 49 §571.201: Occupant protection in interior impact
- CFR Title 49 §571.203: Impact protection for the driver from the steering control system
- CFR Title 49 §571.204: Steering control rearward displacement
- CFR Title 49 §571.206: Door locks and door retention components
- CFR Title 49 §571.207: Seating systems
- CFR Title 49 §571.208: Occupant crash protection
- CFR Title 49 §571.212: Windshield mounting
- CFR Title 49 §571.213: Child restraint systems
- CFR Title 49 §571.214: Side impact protection
- CFR Title 49 §571.216a: Roof crush resistance; Upgraded standard
- CFR Title 49 §571.219: Windshield zone intrusion
- CFR Title 49 §571.223: Rear impact guards
- CFR Title 49 §571.224: Rear impact protection
- CFR Title 49 §571.225: Child restraint anchorage systems

- CFR Title 49 §571.226: Ejection Mitigation
- CFR Title 49 §571.304: Compressed natural gas fuel container integrity
- CFR Title 49 §571.500: Low-speed vehicles
- UNR12: Protection of the driver against the steering mechanism in the event of impact
- UNR13-H: Passenger car braking
- UNR17: Seats, anchorages and any head restraints
- UNR30: Lane Departure Warning Systems
- UNR34: Prevention of fire risks
- UNR46: Devices for indirect vision and their installation in motor vehicles
- UNR66: Strength of superstructure in large passenger vehicles
- UNR87: Daytime running lamps for power-driven vehicles
- UNR152: Advanced Emergency Braking System for M1 and N1 vehicles
- UNR162: Vehicle immobilizers

List of Figures

List of Tables

Bibliography

[BP09] R. Brighi and M. Palmirani. "Legal text analysis of the modification provisions: A pattern oriented approach". In: *Proceedings of the International Conference on Artificial Intelligence and Law*. 2009, pp. 238–239. ISBN: 9781605585970. DOI: 10.1145/1568234.1568272.

[PB10] M. Palmirani and R. Brighi. "Model regularity of legal language in active modifications". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6237 LNAI.M4D (2010), pp. 54–73. ISSN: 03029743. DOI: 10.1007/978-3-642-16524-5_5.

[MRB09] A. Mazzei, D. P. Radicioni, and R. Brighi. "NLP-based extraction of modificatory provisions semantics". In: *Proceedings of the International Conference on Artificial Intelligence and Law*. 2009, pp. 50–57. ISBN: 9781605585970. DOI: 10.1145/1568234.1568241.

[Spi+09] P. L. Spinosa, G. Giardiello, M. Cherubini, S. Marchi, G. Venturi, and S. Montemagni. "NLP-based metadata extraction for legal text consolidation". In: *Proceedings of the International Conference on Artificial Intelligence and Law* (2009), pp. 40–49. DOI: 10.1145/1568234.1568240.

[Les+13] L. Lesmo, A. Mazzei, M. Palmirani, and D. P. Radicioni. *TULSI: An NLP system for extracting legal modificatory provisions*. Vol. 21. 2. 2013, pp. 139–172. ISBN: 1050601291276. DOI: 10.1007/s10506-012-9127-6.

[Mar+03] M. M. Martínez, P. De La Fuente, J. C. Derniame, and A. Pedrero. "Relationship-based dynamic versioning of evolving legal documents". In: *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)* 2543 (2003), pp. 290–305. ISSN: 03029743. DOI: 10.1007/3-540-36524-9_24.

[OIT08] Y. Ogawa, S. Inagaki, and K. Toyama. "Automatic consolidation of Japanese statutes based on formalization of amendment sentences". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4914 LNAI (2008), pp. 363–376. ISSN: 03029743. DOI: 10.1007/978-3-540-78197-4_34.

[XPP16] Xoo, K. Plessas, and A. Plessas. "A semi-automatic system for the consolidation of Greek legislative texts". In: *ACM International Conference Proceeding Series* (2016). DOI: 10.1145/3003733.3003735.

[Lev] L. Levente. *Tauri VS. Electron - Real world application*. URL: https://www.levminer.com/blog/tauri-vs-electron.

[Asa] E. Asaolu. *Tauri vs. Electron: A comparison, how-to, and migration guide*. URL: https://blog.logrocket.com/tauri-electron-comparison-migration-guide/.

[Con] T. Contributers. *Tauri official documentation*. URL: https://tauri.app/v1/guides/building/.

[Fou] O. Foundation. *JSON schema specification*. URL: https://json-schema.org/.