



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

Dipartimento di Scienze

Corso di Laurea in Informatica

# Gestione delle immagini, delle tabelle e dei package in un editor di documenti legali basato su Akoma Ntoso

Relatore:  
Chiar.mo Prof.  
Fabio Vitali

Presentata da:  
Iacopo Porcedda

Correlatore:  
Chiar.ma Prof.ssa  
Monica Palmirani

---

Sessione III

Anno Accademico 2023/2024



# Abstract

Questa tesi descrive l'implementazione di nuove funzionalità nell'editor web Simplex per la redazione di documenti legali in formato Akoma Ntoso. In particolare, il lavoro si è concentrato sulla gestione delle immagini, delle tabelle e sull'implementazione del concetto di package manifestation.

Simplex è stato progettato per superare le limitazioni del suo predecessore LIME, offrendo un'architettura modulare e tecnologie moderne. L'editor si basa su Akoma Ntoso, uno standard internazionale per la rappresentazione strutturata di documenti legislativi, giudiziari e amministrativi.

L'attività di progetto si è focalizzata su tre aspetti principali: la gestione delle immagini, che consente agli utenti di inserire e modificare contenuti grafici; la gestione delle tabelle, che offre strumenti per la creazione e la personalizzazione delle tabelle e del loro stile; e l'implementazione del concetto di package manifestation, che consente di organizzare i componenti del documento in modo conforme allo standard Akoma Ntoso.

I risultati ottenuti mostrano come queste nuove funzionalità abbiano arricchito significativamente le capacità dell'editor, mantenendo al contempo la semplicità d'uso e la conformità allo standard. Il lavoro svolto pone le basi per futuri sviluppi, in particolare per quanto riguarda l'ulteriore estensione delle funzionalità di gestione dei contenuti multimediali e dello stile degli elementi.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Akoma Ntoso</b>	<b>5</b>
2.1	Markup . . . . .	6
2.2	Modello FRBR . . . . .	7
2.3	Schema XML . . . . .	7
2.4	Elementi HTML adottati da Akoma Ntoso . . . . .	8
2.5	Elementi per la gestione delle tabelle . . . . .	10
2.6	Elementi per la gestione delle immagini . . . . .	11
2.7	URI naming convention . . . . .	11
2.8	Componenti . . . . .	13
2.9	Package manifestation . . . . .	14
<b>3</b>	<b>Simplex</b>	<b>17</b>
3.1	LIME . . . . .	17
3.2	Limiti di LIME . . . . .	18
3.3	Da LIME a Simplex . . . . .	19
3.4	Architettura e tecnologie utilizzate . . . . .	20
3.4.1	Frontend . . . . .	20
3.4.2	Backend . . . . .	21
3.5	Interfaccia utente . . . . .	22
3.6	Componenti principali . . . . .	23
<b>4</b>	<b>Gestione di immagini, tabelle e package</b>	<b>29</b>

4.1	Gestione delle immagini . . . . .	29
4.1.1	Inserimento delle immagini . . . . .	30
4.1.2	Modifica delle immagini . . . . .	31
4.1.3	Ridimensionamento delle immagini . . . . .	32
4.1.4	Rimozione delle immagini . . . . .	32
4.1.5	Salvataggio delle immagini . . . . .	32
4.2	Gestione delle tabelle . . . . .	33
4.2.1	Inserimento di tabelle . . . . .	33
4.2.2	Inserimento di righe e colonne . . . . .	34
4.2.3	Rimozione di tabelle, righe e colonne . . . . .	35
4.2.4	Taglia, copia e incolla righe e colonne . . . . .	35
4.2.5	Modifica delle proprietà delle tabelle . . . . .	36
4.2.6	Modifica delle proprietà delle righe . . . . .	37
4.2.7	Modifica delle proprietà delle celle . . . . .	38
4.2.8	Salvataggio delle tabelle . . . . .	39
4.3	Gestione dei package . . . . .	39
<b>5</b>	<b>Implementazione della gestione di immagini, tabelle e package</b>	<b>41</b>
5.1	Implementazione della gestione delle immagini . . . . .	41
5.1.1	Componenti per la gestione delle immagini . . . . .	41
5.1.2	Salvataggio e recupero delle immagini dal server . . . . .	43
5.2	Conversione da HTML ad Akoma Ntoso . . . . .	44
5.2.1	Conversione delle immagini . . . . .	45
5.2.2	Conversione delle tabelle . . . . .	45
5.3	Implementazione della gestione delle tabelle . . . . .	45
5.4	Gestione dello stile . . . . .	49
5.5	Gestione dei package . . . . .	50
<b>6</b>	<b>Valutazione</b>	<b>53</b>
6.1	Test funzionali . . . . .	53
6.1.1	Gestione delle immagini . . . . .	53
6.1.2	Gestione delle tabelle . . . . .	54
6.1.3	Gestione dei package . . . . .	54

6.2	Test di validazione . . . . .	54
6.3	Risultati della valutazione . . . . .	54
<b>7</b>	<b>Conclusioni</b>	<b>57</b>
7.1	Sviluppi futuri . . . . .	58



# Elenco delle figure

2.1	Logo di Akoma Ntoso . . . . .	5
3.1	Architettura di LIME . . . . .	18
3.2	Architettura di Simplex . . . . .	20
3.3	Interfaccia utente di Simplex . . . . .	22
4.1	Menu per l’inserimento delle immagini . . . . .	30
4.2	Menu per la modifica delle immagini . . . . .	31
4.3	Ridimensionamento delle immagini tramite trascinamento . . . . .	32
4.4	Inserimento di tabelle . . . . .	33
4.5	Menu per la gestione delle righe . . . . .	34
4.6	Rimozione della tabella attraverso il menu contestuale . . . . .	35
4.7	Menu delle proprietà delle tabelle . . . . .	36
4.8	Menu delle proprietà delle righe . . . . .	37
4.9	Menu delle proprietà delle celle . . . . .	38
4.10	Esempio di tabella con proprietà modificate . . . . .	39
5.1	Esempio di elemento HTML figure utilizzato nell’editor . . . . .	46
5.2	Esempio di elemento figure dopo la conversione in Akoma Ntoso . . . . .	47
5.3	Esempio di tabella in Akoma Ntoso . . . . .	48



# Capitolo 1

## Introduzione

Nell'era digitale, la gestione e la produzione di documenti legali sta vivendo una profonda trasformazione. La necessità di strumenti efficienti per la creazione, modifica e condivisione di documenti legislativi, giudiziari e amministrativi ha portato allo sviluppo di editor specializzati che devono rispondere a requisiti sempre più complessi.

In questo contesto, lo standard Akoma Ntoso ha assunto un ruolo fondamentale, fornendo un framework per la rappresentazione strutturata dei documenti legali. Nato nel 2004 come progetto delle Nazioni Unite per supportare i parlamenti africani, è oggi riconosciuto come standard internazionale OASIS ed è adottato da numerose istituzioni in tutto il mondo, tra cui il Parlamento Europeo e la Commissione Europea.

Per sfruttare appieno le potenzialità di Akoma Ntoso, è necessario disporre di editor specializzati che ne implementino correttamente le specifiche. LIME è stato uno dei primi editor web sviluppati per questo scopo. Tuttavia, nonostante le sue avanzate funzionalità, LIME presenta alcune limitazioni tecniche che ne rendono difficile l'evoluzione e la manutenzione. In particolare, l'architettura monolitica e l'utilizzo di tecnologie ormai superate hanno reso sempre più complessa l'implementazione di nuove funzionalità.

Per superare queste limitazioni, è nato Simplex, un nuovo editor web progettato con

un'architettura modulare e tecnologie moderne. Questo lavoro di tesi si concentra sull'implementazione di nuove funzionalità chiave in Simplex, in particolare:

- La **gestione delle immagini**, che permette l'inserimento, la modifica e l'organizzazione di contenuti grafici nel documento
- La **gestione delle tabelle**, che permette la creazione e modifica di tabelle, con funzionalità per la personalizzazione di righe, colonne e celle
- L'implementazione del concetto di **package manifestation**, che consente un'organizzazione strutturata dei componenti del documento conforme allo standard Akoma Ntoso

L'implementazione di queste funzionalità mira a fornire agli utenti strumenti intuitivi ed efficienti per la creazione di documenti legali, mantenendo al contempo la piena conformità con lo standard Akoma Ntoso.

La tesi si articola nei seguenti capitoli:

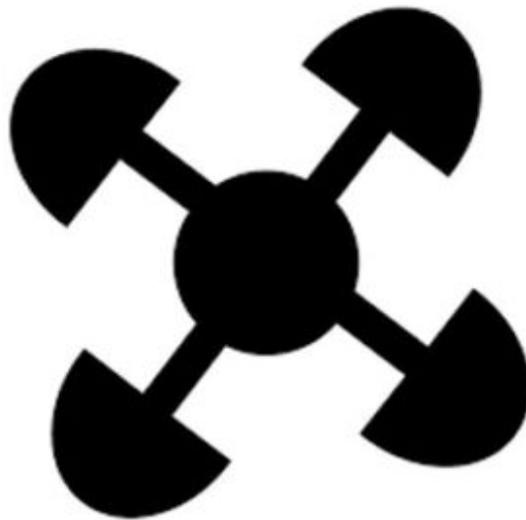
- Il **secondo capitolo** introduce lo standard Akoma Ntoso, analizzandone la storia, gli obiettivi e le caratteristiche principali. Particolare attenzione viene dedicata ai pattern fondamentali dello schema XML e alle specifiche per la gestione di immagini, tabelle e package manifestation.
- Il **terzo capitolo** presenta l'editor Simplex, descrivendone l'architettura, le tecnologie utilizzate e le principali funzionalità. Viene inoltre analizzato il contesto che ha portato al suo sviluppo, partendo dalle limitazioni dell'editor LIME e dalle esigenze degli utenti.
- Il **quarto capitolo** descrive in dettaglio le nuove funzionalità implementate dal punto di vista dell'utente finale. Vengono illustrate le operazioni disponibili per la gestione di immagini e tabelle, nonché le modalità di organizzazione dei documenti attraverso il concetto di package.
- Il **quinto capitolo** si addentra negli aspetti tecnici dell'implementazione delle nuove funzionalità, analizzando le scelte progettuali e le soluzioni adottate. Particolare attenzione viene dedicata alla conversione tra formati HTML e Akoma Ntoso e alla gestione dello stile.

- Il **sesto capitolo** presenta una valutazione del lavoro svolto, analizzando i risultati ottenuti in termini di funzionalità implementate, usabilità dell'editor e conformità allo standard Akoma Ntoso.
- Infine, il **settimo capitolo** raccoglie le conclusioni della tesi, evidenziando i miglioramenti apportati all'editor Simplex e delineando possibili sviluppi futuri.



# Capitolo 2

## Akoma Ntoso



**Figura 2.1:** Logo di Akoma Ntoso

Akoma Ntoso ("cuori uniti" nella lingua Akan dell'Africa occidentale) è uno standard XML per la rappresentazione strutturata di documenti legislativi, giudiziari e amministrativi.

Sviluppato inizialmente nel 2004-2005 come parte di un progetto delle Nazioni Unite per supportare i parlamenti africani, è ora diventato uno standard internazionale

ampiamente adottato.<sup>[VPa]</sup>

Dal 2018 infatti Akoma Ntoso è stato riconosciuto come standard OASIS Open per il modellamento dei documenti giuridici sotto il nome di LegalDocML<sup>[VP18a]</sup>, ed è stato adottato anche dal Parlamento Europeo, dalla Commissione Europea e da numerosi parlamenti in tutto il mondo, tra cui Brasile, Stati Uniti e Italia.<sup>[CPV]</sup>

Gli obiettivi principali di Akoma Ntoso sono:

- Definire un formato comune dei documenti
- Definire un modello comune per lo scambio di documenti
- Definire uno schema dati comune
- Definire uno schema di metadati e un'ontologia comuni
- Definire un modello comune per citazioni e riferimenti incrociati

## 2.1 Markup

Lo standard Akoma Ntoso è stato progettato per rendere i documenti "machine-readable" (leggibili dalle macchine). Questo viene realizzato tramite l'aggiunta di "markup": etichette descrittive associate a tutte le parti del documento. Il markup consente ai computer di analizzare ed elaborare i documenti comprendendo il loro significato e struttura. Il markup può essere categorizzato in strutturale, semantico e presentazionale.

Il **Markup strutturale** definisce la struttura del documento identificando le diverse parti che lo compongono, le loro relazioni e la loro eventuale ripetizione o annidamento all'interno della struttura complessiva.

Il **Markup semantico** arricchisce l'interpretazione semantica delle parti di un documento attraverso etichette contenenti frammenti di testo, come titoli o date.

Il **Markup presentazionale** specifica le istruzioni di formattazione e visualizzazione del contenuto, come le scelte tipografiche e stilistiche applicate a determinati elementi testuali.

## 2.2 Modello FRBR

Akoma Ntoso utilizza il modello FRBR (Functional Requirements for Bibliographic Records). Questo modello descrive quattro livelli sotto i quali una produzione intellettuale, nel nostro caso un documento legale, può essere vista:

- **Work:** Il concetto astratto della risorsa legale, ad esempio la legge italiana 137/2004.
- **Expression:** Qualsiasi versione del work il cui contenuto è specificato e differente dalle altre per qualsiasi motivo, come la lingua. Ad esempio l'attuale versione consolidata della legge Italiana 137/2004.
- **Manifestation:** Qualsiasi formato fisico o digitale dell'expression, ad esempio una rappresentazione in XML dell'attuale versione consolidata della legge Italiana 137/2004
- **Item:** La copia fisica di qualsiasi manifestation, intesa come file archiviato su un computer, connesso o meno alla rete. Ad esempio il file "legge137-2004.xml" posto in una specifica cartella su uno specifico computer.

## 2.3 Schema XML

Lo schema XML di Akoma Ntoso si fonda su sei pattern che stabiliscono il content model per gli elementi dello standard. Per ogni pattern viene fornito un elemento generico che rispetta il content model del pattern. L'elemento può essere specificato assegnandogli un nome non utilizzato nello schema attraverso l'attributo "name". Di seguito vengono descritti i pattern e le loro caratteristiche.

### Marker

I marker sono elementi privi di contenuto che acquisiscono significato in base alla loro posizione nel testo, al loro nome e ai loro attributi. Esistono due principali famiglie di marker: i placeholder, che possono apparire in qualsiasi punto in cui può essere presente un testo, e gli elementi per i metadati, che possono essere presenti solo nella sezione dedicata ai metadati.

## **Inline**

Un elemento inline è un contenitore di testo che può essere inserito all'interno di un elemento con pattern block. Questo pattern è utilizzato per indicare caratteristiche semantiche o strutturali di una porzione di testo.

## **Block**

Il pattern block indica un contenitore di testo indipendente dagli altri elementi. Gli elementi block possono contenere elementi inline.

## **Container**

Un container è un elemento strutturale che raggruppa altri elementi e assegna loro un nome cumulativo. Esistono varie categorie di container, e ognuna può contenere elementi di tipo diverso. Non esiste un singolo content model, ma diversi content model per ogni categoria.

## **Hierarchical container (hcontainer)**

Un hcontainer è un tipo speciale di container che organizza il contenuto in una gerarchia di contenimento. Una gerarchia è una struttura a livelli nidificati, ciascuno con un titolo e una numerazione. Ogni livello può includere ulteriori sezioni nidificate o blocchi di testo. Il testo non può essere inserito direttamente nella gerarchia, ma deve essere contenuto in elementi block appropriati.

## **Popup**

I popup sono elementi che possono essere inseriti all'interno di un flusso di testo e di elementi inline. Questi elementi indicano una struttura indipendente che non interagisce con il testo e con gli elementi inline che li circondano.

## **2.4 Elementi HTML adottati da Akoma Ntoso**

Lo schema di Akoma Ntoso contiene alcuni elementi adottati da HTML il cui scopo è principalmente orientato alla presentazione. Questi elementi sono stati inseriti nel

namespace di Akoma Ntoso, e dunque non appartengono al namespace HTML. In alcuni casi, la loro semantica coincide esattamente, come nel caso di "span", mentre in altri differisce, come per "div".

Questi elementi vengono divisi in quattro categorie: HTMLcontainers, HTMLblock, HTMLinline e HTMLmarker.

### **HTMLcontainers**

L'unico elemento appartenente a questa categoria è "div". A differenza di HTML, questo elemento non viene usato come blocco generico, ma come elemento container generico.

### **HTMLblock**

L'elemento "p" è usato in Akoma Ntoso come blocco generico. Gli elementi "ol" e "ul", come in HTML, vengono usati per definire liste numerate e non numerate. L'elemento "li" è usato per definire i singoli elementi della lista. Gli elementi "table", "tr", "td", "th" e "caption" vengono usati per definire le tabelle.

### **HTMLinline**

Gli elementi "b", "i", "a", "u", "sub", "sup" e "abbr" sono definiti in Akoma Ntoso come elementi HTMLinline e mantengono la loro semantica HTML.

- **b**: testo in grassetto
- **i**: testo in corsivo
- **a**: link ipertestuale
- **u**: testo sottolineato
- **sub**: testo in pedice
- **sup**: testo in apice
- **abbr**: abbreviazione

## HTMLmarker

Gli elementi appartenenti a questa categoria sono "br" e "img". L'elemento "br" è utilizzato per inserire un'interruzione di riga mentre l'elemento "img" è utilizzato per inserire un'immagine.

## Attributi HTML adottati da Akoma Ntoso

Akoma Ntoso definisce un insieme di attributi principali, chiamati "core", che possono essere applicati a qualsiasi elemento presente nella parte di contenuto del documento. All'interno di questo insieme è presente la categoria "HTMLattrs", che include gli attributi HTML "class", "style" e "title". Questi attributi permettono di definire stile, classi e titolo di un elemento in modo analogo a HTML.

## 2.5 Elementi per la gestione delle tabelle

In questa sezione analizziamo in dettaglio gli elementi di Akoma Ntoso utilizzati per implementare la gestione delle tabelle nell'editor Simplex.

Analogamente ad HTML, l'elemento "table", "tr", "td" e "th" permettono la definizione delle tabelle in Akoma Ntoso. Questi elementi fanno parte del gruppo "HTMLblock".

L'elemento "table" rappresenta il contenitore principale della tabella. La sua struttura può includere un elemento "caption" opzionale per la didascalia e uno o più elementi "tr" che definiscono le righe. Oltre agli attributi core di Akoma Ntoso, "table" supporta attributi specifici per il controllo della presentazione: "width" per definire la larghezza della tabella, "border" per lo spessore e stile dei bordi, "cell-spacing" per la spaziatura tra le celle e "cellpadding" per la spaziatura interna alle celle.

L'elemento "caption" è un elemento di tipo inline, utilizzato per definire la didascalia di una tabella. Questo elemento rispetta le proprietà di un elemento inline di Akoma Ntoso e non contiene attributi specifici.

L'elemento "tr" è usato per definire una riga di una tabella. Può avere come figli elementi "td" o "th" e non può contenere attributi specifici oltre a quelli principali di Akoma Ntoso.

Gli elementi "td" (table data) e "th" (table header) rappresentano rispettivamente le celle normali e le celle di intestazione. Supportano gli attributi specifici "rowspan" e "colspan" per controllare l'espansione delle celle su più righe o colonne.

## 2.6 Elementi per la gestione delle immagini

L'elemento "img", appartenente al gruppo "HTMLmarker", funge da puntatore all'interno della documento XML a livello manifestation, indicando il punto in cui l'immagine deve essere inserita nel documento.

Gli attributi "src" e "alt" vengono utilizzati in riferimenti a livello manifestation. L'attributo "src" contiene l'IRI di livello manifestation della risorsa da inserire, mentre l'attributo "alt" contiene il testo da mostrare in caso il caricamento della risorsa non sia possibile. Inoltre, gli attributi "width" e "height" permettono di definire rispettivamente larghezza e altezza dell'immagine nel documento.<sup>[VP18b]</sup>

A differenza di HTML, che definisce diversi elementi per la gestione delle immagini come "figure" e "figcaption", Akoma Ntoso definisce unicamente l'elemento "img". Questo significa che quando l'editor Simplex utilizza elementi come "figure" e "figcaption", questi devono essere convertiti rispettivamente in elementi "div" (container generico) e "p" (blocco generico) poiché non sono definiti nello schema Akoma Ntoso.

## 2.7 URI naming convention

Oltre al vocabolario XML, Akoma Ntoso definisce una convenzione per la denominazione degli URI (URI naming convention) che rappresenta una componente fondamentale dello standard.

Per comprendere l'URI naming convention di Akoma Ntoso, è necessario prima chiarire alcuni concetti chiave. Gli URI (Uniform Resource Identifiers) sono identi-

ficatori standardizzati utilizzati per riferirsi univocamente a risorse sul Web, come documenti, servizi o concetti. Gli IRI (International Resource Identifier) sono una versione estesa degli URI che supporta caratteri Unicode non-ASCII, permettendo l'uso di alfabeti non latini. Gli URL (Uniform Resource Locators) sono un tipo specifico di URI/IRI che indica la posizione fisica di una risorsa, specificando il protocollo di accesso, il server e il percorso. Un'URI naming convention è un insieme di regole e linee guida per assegnare nomi coerenti agli URI all'interno di un sistema.

La naming convention di Akoma Ntoso è stata progettata per garantire la massima flessibilità e indipendenza dall'architettura di archiviazione sottostante. Un aspetto chiave è la distinzione tra gli IRI logici memorizzati nei documenti e gli URL fisici che puntano alle effettive risorse. Questa separazione permette di modificare la struttura fisica dell'archivio senza impattare i riferimenti logici contenuti nei documenti.

La naming convention si basa su una struttura gerarchica che riflette il modello FRBR (Functional Requirements for Bibliographic Records). Un IRI Akoma Ntoso è composto da quattro sezioni principali:

```
[naming authority]//[workfeatures]/[expressionfeatures]/[manifestation...  
...features]/[itemfeatures]
```

Le informazioni contenute in un IRI sono divise in "features" per ogni livello della gerarchia FRBR e separate dal carattere "/". Di seguito vengono riportate le features per ogni livello.

Features del livello work:

- Il prefisso "/akn"
- Il codice del paese o suddivisione
- Il tipo e sottotipo di documento
- Il soggetto emanante
- Data di creazione
- Numero o titolo del work per disambiguazione

Features del livello expression:

- Il codice della lingua
- Il carattere ”@” seguito dall’identificatore di versione
- Date specifiche del contenuto (opzionale)
- Informazioni di authoring del contenuto (opzionale)

Features del livello manifestation:

- Informazioni di authoring del markup (opzionale)
- Date specifiche del markup (opzionale)
- Informazioni sulle annotazioni del markup (opzionale)
- Il carattere ”.” seguito dal formato del documento

Le features del livello Item non sono definite nella naming convention in quanto Akoma Ntoso non fa alcuna assunzione sul meccanismo di archiviazione fisica delle manifestation.<sup>[VPb][VP19]</sup>

## 2.8 Componenti

In Akoma Ntoso, un componente rappresenta una parte distinta e autonoma di un documento che può essere identificata e referenziata in modo indipendente. I componenti sono tipicamente definiti a livello expression, poiché sono un modo per organizzare il contenuto.

I componenti possono essere referenziati anche a livello work quando è necessario mantenere riferimenti stabili nel tempo, indipendentemente da eventuali modifiche alla loro posizione o denominazione. Ad esempio, se un documento contiene un riferimento alla ”tabella A” presente nell’allegato 1, e successivamente l’allegato 1 viene abrogato, la tabella A diventerà implicitamente un allegato del documento principale. Dopo questo evento, i riferimenti originali alla tabella A dell’allegato 1 sono considerati come riferimenti alla tabella A nel documento principale.

La struttura dell’IRI di un componente è costruita concatenando l’IRI del document a livello work, expression o manifestation di riferimento, seguito dai caratteri ”/”

e "!" e infine il nome del componente. Il nome "main" è riservato al componente principale o quando il documento ha un solo componente. Altri componenti possono avere nomi arbitrari univoci, opzionalmente seguiti da un numero identificativo. Nel caso di componenti a livello manifestation, l'estensione del file verrà spostata alla fine dell'IRI.

Ad esempio, un IRI di componente a livello expression potrebbe essere:

[<http://www.authority.org/akn/dz/minutes/2004-12-21/nn/fra/!main>]

## 2.9 Package manifestation

Vediamo ora in dettaglio il package manifestation, un concetto chiave nel lavoro di progetto svolto sull'editor Simplex.

Un package manifestation è un package composto da uno o più file memorizzati allo stesso livello. Il package viene salvato e trasportato come file zip con estensione ".akn". Può contenere file con formato XML ma anche altri formati multimediali se necessario.

In particolare, un package deve rispettare le seguenti regole:

- Se il documento è composto solo da testo e non contiene riferimenti a nessun componente multimediale di alcun tipo, allora il package conterrà un singolo documento chiamato "main.xml".
- Se il documento è composto da vari componenti a livello manifestation ma non contiene nessun riferimento a componenti multimediali, allora il package sarà composto da diversi file XML, uno per ciascun componente a livello expression. Ogni componente a livello manifestation avrà lo stesso nome del componente a livello expression, più l'estensione ".xml". Il nome "main" è riservato al componente principale. I numeri vengono utilizzati solamente se fanno già parte del componente a livello expression.
- Se il documento include riferimenti a componenti multimediali, questi devono essere memorizzati all'interno del package in una cartella denominata "media". I file multimediali possono avere nomi arbitrari, ma devono avere l'e-

stensione corretta per il loro tipo di contenuto. Ad esempio, un'immagine PNG deve avere l'estensione ".png". I componenti multimediali non hanno un corrispondente componente a livello expression, ma esistono solo a livello di manifestation e sono specificati negli attributi "src", ad esempio negli elementi "img".

Tutti i riferimenti tra i componenti a livello manifestation devono essere relativi e interni al package. Ad esempio, il componente principale, "main.xml", si riferirà agli altri componenti a livello manifestation tramite elementi "attachment" utilizzando riferimenti relativi, come "attachment1.xml". Allo stesso modo, un componente potrà riferirsi all'immagine "image1" utilizzando il percorso relativo "media/image1.png".<sup>[VP19]</sup>



# Capitolo 3

## Simplex

Simplex è un editor basato su Akoma Ntoso per la stesura e il markup di documenti legali. È stato sviluppato per rispondere alla necessità di migliorare e ampliare le funzionalità dell'editor LIME, che presentava diverse problematiche.

### 3.1 LIME

LIME è un editor web altamente personalizzabile progettato per guidare l'utente nella trasformazione di documenti non strutturati in documenti XML strutturati e ben formati, con la possibilità di renderli validi secondo il plugin linguistico scelto dall'utente. Sviluppato dal CIRSIFID e dall'Università di Bologna, questo editor è open source ed è basato su tecnologie open source.<sup>[CPCdB]</sup>

L'interfaccia utente dell'applicazione può essere suddivisa nelle seguenti sezioni:

- **Barra degli strumenti:** Si trova nella parte superiore e consente all'utente di accedere alle funzionalità principali, come l'importazione e l'esportazione di documenti.
- **Pannello outline:** Posizionato a sinistra, permette all'utente di vedere la struttura del documento.
- **Editor testuale:** La parte centrale dell'interfaccia. Consente di visualizzare e modificare il documento.

- **Pannello di markup:** Si trova nella parte destra e si occupa dell'aggiunta di elementi Akoma Ntoso al documento.
- **Editor di metadati:** Posizionato nella parte inferiore. Consente la visualizzazione, l'inserimento, la modifica e la rimozione dei metadati all'interno del documento.

## 3.2 Limiti di LIME

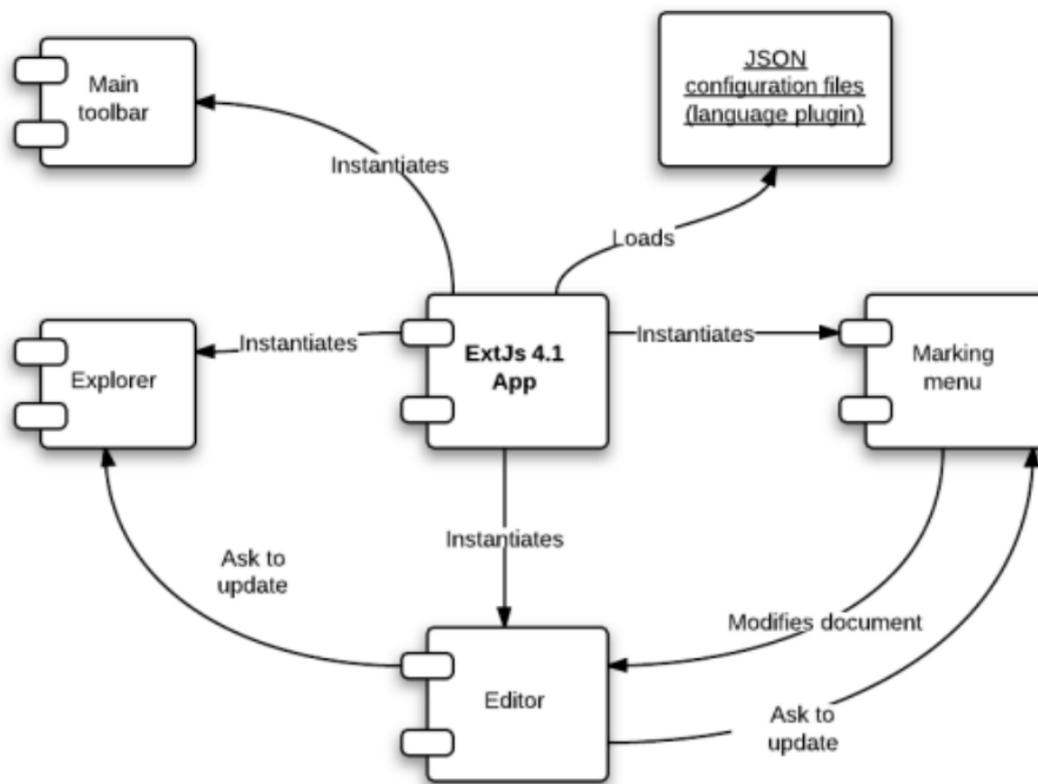


Figura 3.1: Architettura di LIME

LIME presenta diversi limiti architetturali che ne compromettono la flessibilità.

L'adozione di un'architettura monolitica rende ogni componente strettamente interconnesso e dipendente dagli altri, causando difficoltà nell'aggiunta di nuove funzio-

nalità e nella risoluzione dei bug. Inoltre, la code base continua a crescere, rendendo sempre più difficile la comprensione del codice.

Inoltre, l'utilizzo di tecnologie datate, come il framework ExtJS, ha comportato ulteriori problematiche. Le dimensioni considerevoli della libreria ExtJS hanno provocato rallentamenti significativi nel caricamento dell'applicazione, e il framework stesso risulta difficile da apprendere.

### 3.3 Da LIME a Simplex

Per far fronte ai problemi legati a LIME, si è deciso di sviluppare un nuovo editor, che prende il nome di Simplex.

Simplex offre le stesse funzionalità di LIME, garantendo maggiore modularità, flessibilità e la piena conformità allo standard Akoma Ntoso.

A differenza di LIME, Simplex è stato progettato con un'architettura modulare e scalabile. Questo approccio consente di sviluppare nuovi componenti, come l'editor delle immagini o delle tabelle di cui mi sono occupato, senza la necessità di conoscere in dettaglio l'intera code base. Ogni componente è indipendente dagli altri, rendendo lo sviluppo e l'integrazione di nuove funzionalità più rapidi ed efficienti.

Un altro punto di forza di Simplex, che lo rende molto più flessibile rispetto al suo predecessore, è la possibilità di definire un file di configurazione che descrive l'editor sia a livello grafico che funzionale. È possibile ottenere versioni distinte di Simplex semplicemente modificando i valori di questo file, senza dover cambiare il codice dell'applicazione.<sup>[Dzi22]</sup>

## 3.4 Architettura e tecnologie utilizzate

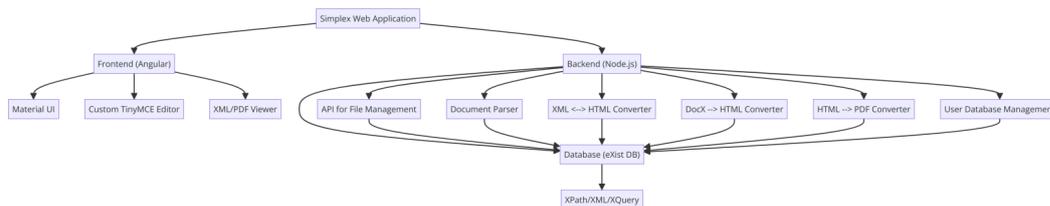


Figura 3.2: Architettura di Simplex

L'architettura di Simplex può essere suddivisa, come la maggior parte delle applicazioni web, in due parti principali: il backend e il frontend.

### 3.4.1 Frontend

Per il frontend è stato utilizzato il framework Angular, una piattaforma open-source basata su JavaScript e sviluppata da Google. Questo framework semplifica lo sviluppo di applicazioni web dinamiche e reattive grazie alla sua architettura basata su componenti. Ogni componente in Angular è una parte riutilizzabile dell'interfaccia utente, che gestisce la propria logica e il proprio stato. Angular permette la definizione di Service, classi TypeScript che forniscono funzionalità condivise tra i componenti. L'utilizzo di una struttura modulare consente di costruire applicazioni complesse in modo più organizzato e mantenibile, facilitando la gestione e l'aggiornamento del codice.

In aggiunta, è stato utilizzato Material UI, una libreria di componenti per l'interfaccia utente che segue le linee guida del Material Design di Google. Material UI fornisce una serie di componenti predefiniti e stilizzati che aiutano a creare interfacce utente moderne e coerenti.

Un'altra libreria fondamentale è quella per l'editing del documento. Come per LIME, è stato scelto TinyMCE, un editor WYSIWYG (What You See Is What You Get) che consente di modificare i contenuti del documento attraverso un'interfaccia grafica, senza dover scrivere codice HTML e CSS. L'editor è stato adattato agli elementi e regole dello schema di Akoma Ntoso, così da generare documenti che rispettino

lo standard. Questa libreria è stata essenziale nella realizzazione dei componenti relativi a immagini e tabelle.

### 3.4.2 Backend

Il backend di Simplex può essere suddiviso in due parti principali: il server e il database.

Il server utilizza Node.js, un ambiente di runtime open-source che permette l'esecuzione di codice JavaScript al di fuori di un browser. Il server si occupa della gestione dei documenti, e interagisce con il database per salvare e recuperare i documenti. Un altro componente fondamentale è il convertitore di documenti, che offre le seguenti conversioni:

- Da XML a HTML e viceversa
- Da DocX a HTML
- Da HTML a PDF
- Da AKN a PDF

Il sistema di persistenza dei dati di Simplex si basa su due database distinti. Per la gestione dei documenti viene utilizzato eXist-db, un database nativo XML open-source specializzato nella memorizzazione, gestione e interrogazione di documenti XML. La scelta di eXist-db è stata dettata dalla sua naturale compatibilità con il formato XML utilizzato da Akoma Ntoso. Per quanto riguarda invece la gestione degli utenti e delle loro informazioni, viene impiegato MongoDB, un database NoSQL open-source che memorizza i dati in formato BSON (Binary JSON), garantendo flessibilità e scalabilità nella gestione dei profili utente.

## 3.5 Interfaccia utente

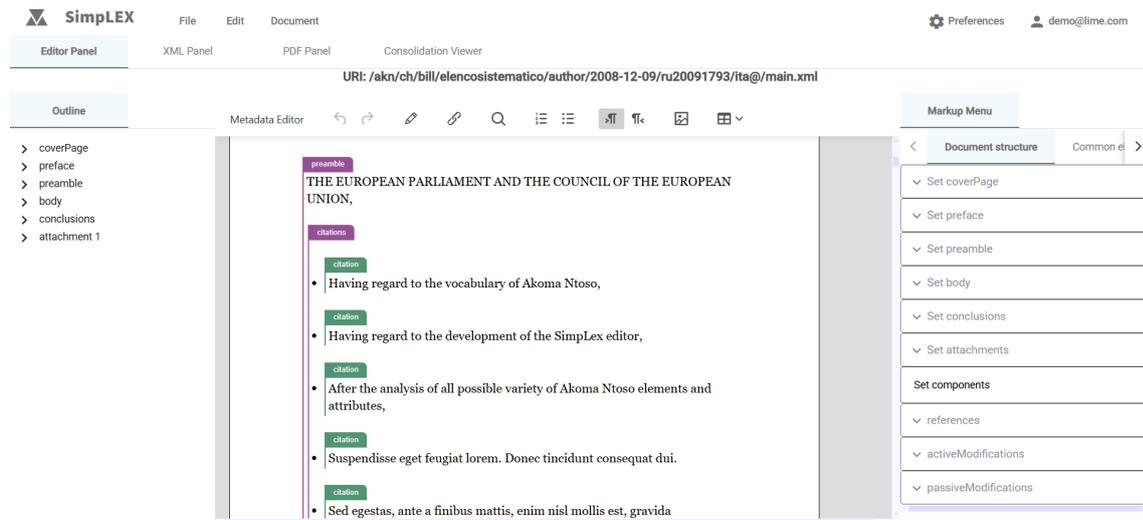


Figura 3.3: Interfaccia utente di Simplex

L'interfaccia utente di Simplex è composta da diversi componenti Angular, ognuno responsabile di specifiche funzionalità dell'editor. Questi componenti interagiscono tra loro attraverso funzioni e Service Angular per garantire il corretto funzionamento dell'applicazione.

Il componente centrale è l'AknTinyMceIntegrator, che funge da intermediario tra l'interfaccia grafica e i servizi dell'applicazione. Questo componente segue il pattern Model-View-Controller (MVC):

- Come Controller, gestisce le azioni dell'utente
- Coordina il Model, organizzando e processando i dati
- Aggiorna la View, mantenendo sincronizzata l'interfaccia utente<sup>[Pog24]</sup>

A differenza di LIME, l'interfaccia utente di Simplex è completamente personalizzabile attraverso un file di configurazione. Attraverso il campo "layout" è possibile definire l'organizzazione e la disposizione dei vari elementi nell'interfaccia, mentre il campo "components" permette di definire nuovi componenti da aggiungere. Grazie a questo approccio configurativo e all'architettura modulare di Simplex, è possibile

estendere facilmente le funzionalità dell'editor aggiungendo nuovi componenti, come nel caso degli editor specializzati per la gestione di immagini e tabelle.

## 3.6 Componenti principali

Nelle sezioni successive verranno presentati i componenti secondo la configurazione standard di Simplex, che come detto in precedenza può essere completamente personalizzata in base alle necessità dell'utente attraverso il file di configurazione.

### Header

L'header è posizionato nella parte superiore dell'interfaccia e ospita la barra degli strumenti principale, che fornisce accesso ai seguenti menu:

- **File:** Gestisce le operazioni fondamentali sui documenti, incluse la creazione, l'apertura e il salvataggio. Offre inoltre funzionalità di import/export nei formati AKN, HTML e PDF.
- **Edit:** Raccoglie alcuni strumenti di editing dei documenti, tra cui il markup automatico e la generazione della tabella dei contenuti.
- **Document:** Fornisce ulteriori strumenti per la gestione dei documenti, come la validazione XML.
- **Preferences:** Permette di configurare alcune opzioni dell'editor come lingua, visibilità dei pannelli e modalità di visualizzazione.
- **Account:** Si occupa della gestione del profilo utente, offrendo accesso alle impostazioni personali e alla funzionalità di logout.

### Tab container

Il tab container si trova sotto l'header e permette la selezione del pannello visualizzato nella parte principale dell'interfaccia. I pannelli selezionabili sono:

- **Editor Panel:** Il pannello principale che permette la visualizzazione e modifica del documento. Questo pannello contiene la maggior parte dei componenti che verranno presentati nelle prossime sezioni.

- **XML Panel:** Permette di visualizzare il documento in formato XML.
- **PDF Panel:** Permette di visualizzare il documento in formato PDF.
- **Consolidation Viewer:** Strumento per il consolidamento dei documenti legislativi. Permette di integrare in un solo documento l'atto originale e tutte le sue successive modifiche.<sup>[Fer24]</sup>

## Barra degli strumenti

La barra degli strumenti dell'editor, implementata attraverso la libreria TinyMCE, offre accesso a tutte le operazioni disponibili per la modifica del documento attualmente visualizzato. Gli strumenti disponibili includono:

- **Metadata Editor:** Consente l'accesso all'editor dei metadati, spiegato in dettaglio nella sezione 3.6.
- **Undo/Redo:** Consentono di annullare o ripristinare le modifiche recenti.
- **Track Changes:** Attiva il tracciamento delle modifiche, evidenziando le aggiunte ed eliminazioni effettuate nel testo.
- **Find and Replace:** Consente la ricerca e sostituzione del testo all'interno del documento.
- **Reference insertion:** Permette l'inserimento di riferimenti.
- **Numbered e bullet list:** Permettono l'inserimento di elenchi numerati e puntati.
- **Right to left e Left to right:** Consentono di invertire la direzione del testo.
- **Insert/edit image:** Permette di inserire e modificare immagini. Maggiori dettagli sono disponibili nella sezione 4.1.
- **Table:** Offre funzionalità per inserire e modificare tabelle. Maggiori dettagli sono disponibili nella sezione 4.2.

## **Pannello outline**

Il pannello outline, posizionato nella parte sinistra dell'interfaccia, fornisce una rappresentazione gerarchica del documento Akoma Ntoso in forma di albero. Ogni elemento della struttura è rappresentato con il proprio nome e mostra i relativi elementi figli, offrendo così una chiara panoramica dell'intero documento.

Oltre alla funzione di visualizzazione, il pannello outline serve anche come strumento di navigazione. Ogni elemento mostrato nell'albero funziona come collegamento ipertestuale: cliccando su di esso, l'utente viene immediatamente portato alla sezione corrispondente nel documento. Questa caratteristica permette una navigazione rapida attraverso le diverse sezioni del documento, semplificando notevolmente l'esplorazione di documenti lunghi o complessi.

## **Editor dei documenti**

L'editor dei documenti, posizionato nella parte centrale dell'interfaccia, rappresenta il componente principale dell'applicazione. Implementato tramite la libreria TinyMCE, questo componente consente agli utenti di visualizzare e modificare direttamente il contenuto del documento.

L'interfaccia riprende le caratteristiche di un comune editor di testo, arricchendole con il supporto allo standard Akoma Ntoso. Durante la stesura, il documento viene automaticamente validato secondo le regole dello standard, mentre i vari pattern Akoma Ntoso vengono evidenziati attraverso etichette colorate per una migliore comprensione della struttura.

## **Pannello di markup**

Il pannello di markup, posizionato nella parte destra dell'interfaccia, permette la marcatura dei documenti secondo lo schema Akoma Ntoso. Questo pannello presenta un'interfaccia composta da pulsanti interattivi, organizzati secondo una struttura gerarchica a più livelli. L'organizzazione dei pulsanti segue una disposizione gerarchica, dove ogni livello mostra gli elementi che possono essere inseriti come figli dell'elemento superiore. Questa organizzazione aiuta l'utente a mantenere una corretta struttura del documento durante il processo di marcatura.

Il pannello si divide in due sezioni principali:

- **Document structure:** raccoglie tutti gli elementi disponibili per la particolare giurisdizione e tipologia del documento in uso.
- **Common elements:** contiene elementi generici utilizzabili in qualsiasi contesto, particolarmente utili quando non sono disponibili elementi specifici più appropriati.

### Editor di metadati

L'editor dei metadati può essere attivato attraverso il pulsante "Metadata Editor" nella barra degli strumenti. Questo strumento permette di visualizzare, modificare e arricchire i metadati rispettando lo standard Akoma Ntoso.

L'editor dei metadati è organizzato in diverse categorie, basate sui diversi tag per metadati utilizzati in Akoma Ntoso:

- **Document:** Raccoglie le informazioni del documento come data di creazione, autore e lingua. Questi metadati sono contenuti nel tag "identification" dello schema Akoma Ntoso.
- **Publication:** Contiene i dettagli relativi alla pubblicazione del documento contenuti nel tag "publication".
- **Events:** Traccia la cronologia degli eventi che hanno interessato il documento legale. Si riferisce al tag "lifecycle".
- **Workflow:** Documenta le fasi del processo legislativo o giuridico del documento. Modifica il contenuto del tag "workflow".
- **Classification:** Permette di categorizzare il documento o suoi frammenti attraverso parole chiave basate sul contenuto legale, utilizzando il tag "classification".
- **References:** Contiene i riferimenti ad altri documenti o sezioni. Fa riferimento al tag "references".<sup>[Zec22]</sup>

## Menu contestuale

Il menu contestuale rappresenta uno strumento fondamentale per la gestione e modifica della struttura del documento. Offre all'utente una serie di funzionalità intuitive e rapide, accessibili con un semplice click del tasto destro del mouse. Le principali operazioni disponibili nel menu contestuale sono:

- **Mark:** Permette di applicare rapidamente markup al testo selezionato. Mostra una lista di elementi validi che possono essere inseriti all'interno del pattern dell'elemento corrente.
- **Unmark:** Consente di rimuovere il markup da un elemento. Offre un modo veloce per eliminare etichette inserite erroneamente.
- **Change to:** Permette di modificare il pattern di un elemento esistente. Mostra solo le alternative valide compatibili con il pattern dell'elemento corrente, evitando la necessità di rimuovere e riapplicare il markup.
- **Delete:** Permette di rimuovere l'elemento selezionato.

Quando l'utente seleziona una tabella, viene mostrato un ulteriore menu contestuale che offre un accesso rapido alle operazioni principali della tabella. Per una descrizione dettagliata di tutte le operazioni disponibili, si rimanda alla sezione 4.2.



# Capitolo 4

## Gestione di immagini, tabelle e package

Questo capitolo illustra le nuove funzionalità implementate nell'editor Simplex, presentandole dal punto di vista dell'utente finale. In particolare, verranno descritte in dettaglio le funzionalità relative alla gestione di immagini e tabelle, oltre all'integrazione del concetto di package manifestation. Per maggiori dettagli sull'implementazione si rimanda al capitolo 5.

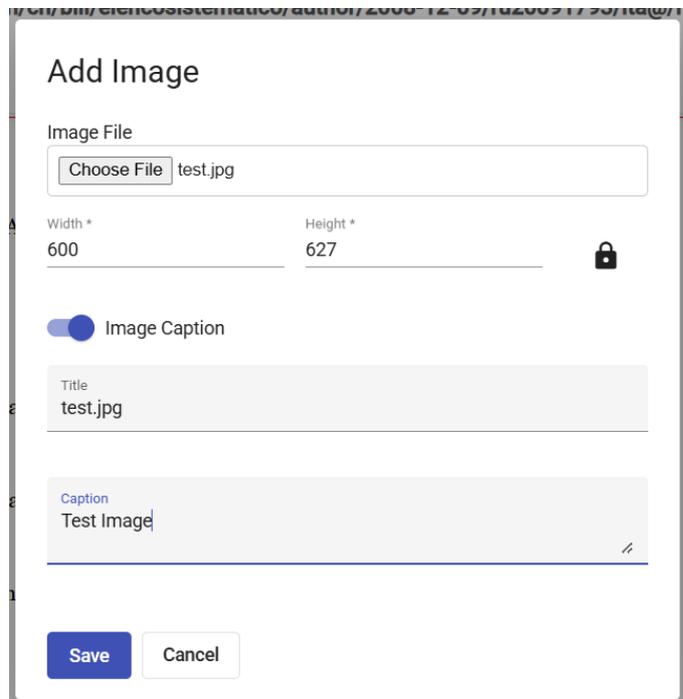
### 4.1 Gestione delle immagini

La gestione delle immagini in Simplex offre un insieme completo di funzionalità che permettono agli utenti di inserire e modificare immagini all'interno dei documenti. Le principali operazioni disponibili sono:

- Inserimento di immagini
- Modifica delle immagini
- Ridimensionamento delle immagini
- Rimozione delle immagini
- Salvataggio delle immagini

Di seguito verranno illustrate in dettaglio queste funzionalità.

### 4.1.1 Inserimento delle immagini



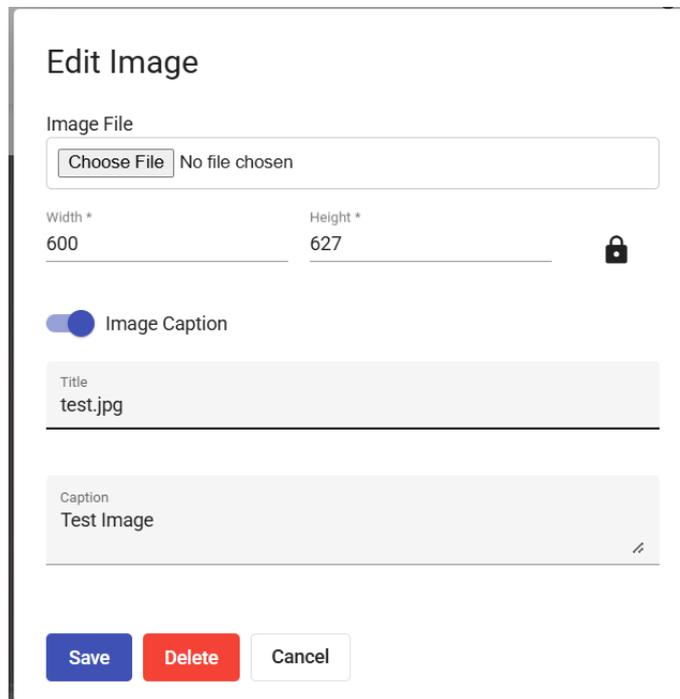
**Figura 4.1:** Menu per l’inserimento delle immagini

L’inserimento di immagini avviene tramite l’apposito pulsante ”Add/Edit Image” nella barra degli strumenti. L’utente può aggiungere una nuova immagine selezionandola dal proprio dispositivo o trascinandola direttamente nell’editor. Una volta selezionata l’immagine, è possibile personalizzarne le proprietà attraverso un menu dedicato. In particolare, si possono modificare:

- **Dimensioni:** larghezza e altezza dell’immagine, con la possibilità di mantenere bloccato l’aspect ratio o modificare liberamente le proporzioni
- **Titolo:** nome identificativo dell’immagine mostrato come suggerimento quando si passa il mouse sopra l’immagine
- **Didascalia:** testo descrittivo che verrà visualizzato sotto l’immagine nel documento

L'inserimento dell'immagine può essere confermato tramite il pulsante "Save", oppure annullato con il pulsante "Cancel".

### 4.1.2 Modifica delle immagini

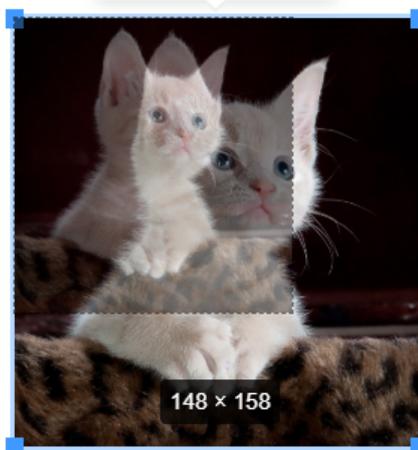


The image shows a dialog box titled "Edit Image". At the top, there is a section labeled "Image File" with a "Choose File" button and the text "No file chosen". Below this are two input fields: "Width \*" with the value "600" and "Height \*" with the value "627". To the right of these fields is a lock icon. Underneath is a toggle switch labeled "Image Caption" which is currently turned on. Below the toggle are two text input fields: "Title" containing "test.jpg" and "Caption" containing "Test Image". At the bottom of the dialog are three buttons: "Save" (blue), "Delete" (red), and "Cancel" (white).

**Figura 4.2:** Menu per la modifica delle immagini

Per modificare un'immagine esistente, è sufficiente selezionarla e premere il pulsante "Add/Edit Image". Il pulsante apparirà in uno stato attivo, segnalando che è possibile modificare l'immagine selezionata. Cliccando sul pulsante si aprirà il menu contenente tutte le proprietà attuali dell'immagine, come dimensioni, titolo e didascalia. L'utente può modificare questi valori secondo le proprie esigenze. Le modifiche possono essere confermate tramite il pulsante "Save", che aggiornerà l'immagine nel documento con le nuove proprietà, oppure annullate tramite il pulsante "Cancel", che ripristinerà i valori originali.

### 4.1.3 Ridimensionamento delle immagini



Test Image

**Figura 4.3:** Ridimensionamento delle immagini tramite trascinamento

Le immagini possono essere ridimensionate non solo attraverso il menu di modifica, ma anche tramite trascinamento. Selezionando un'immagine, è possibile trascinare i bordi per modificarne le dimensioni: trascinando verso l'alto o il basso si regola l'altezza, mentre trascinando verso destra o sinistra si modifica la larghezza. Questa modalità di ridimensionamento offre un controllo più intuitivo e immediato rispetto all'inserimento manuale delle dimensioni nel menu.

### 4.1.4 Rimozione delle immagini

Per rimuovere un'immagine dal documento è possibile utilizzare il pulsante "Remove" presente nel menu di modifica. Questa operazione rimuoverà completamente l'immagine selezionata dal documento, inclusa la sua didascalia.

### 4.1.5 Salvataggio delle immagini

Le immagini inserite o modificate vengono salvate automaticamente insieme al resto del documento quando viene effettuato il salvataggio tramite il pulsante "File > Save".

## 4.2 Gestione delle tabelle

L'editor fornisce un insieme completo di strumenti per la gestione delle tabelle, che permettono di creare e modificare tabelle in modo intuitivo. Le principali operazioni disponibili sono:

- Inserimento di tabelle
- Inserimento di righe e colonne
- Rimozione di tabelle, righe e colonne
- Taglia, copia e incolla righe e colonne
- Modifica delle proprietà delle tabelle
- Modifica delle proprietà delle righe
- Modifica delle proprietà delle celle
- Salvataggio delle tabelle

Di seguito verranno illustrate in dettaglio queste funzionalità.

### 4.2.1 Inserimento di tabelle

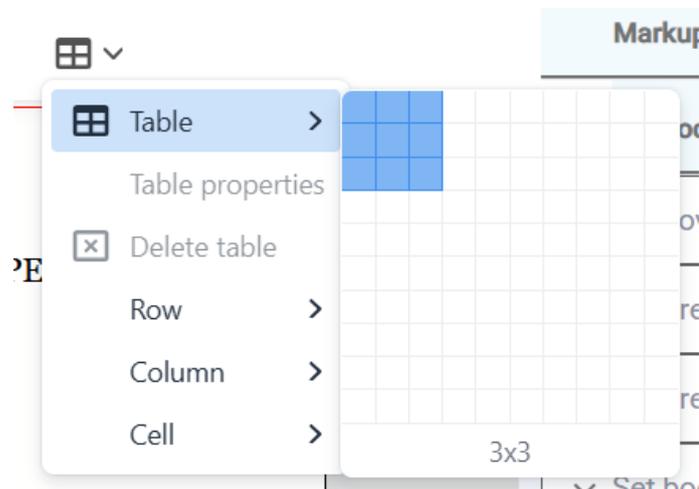
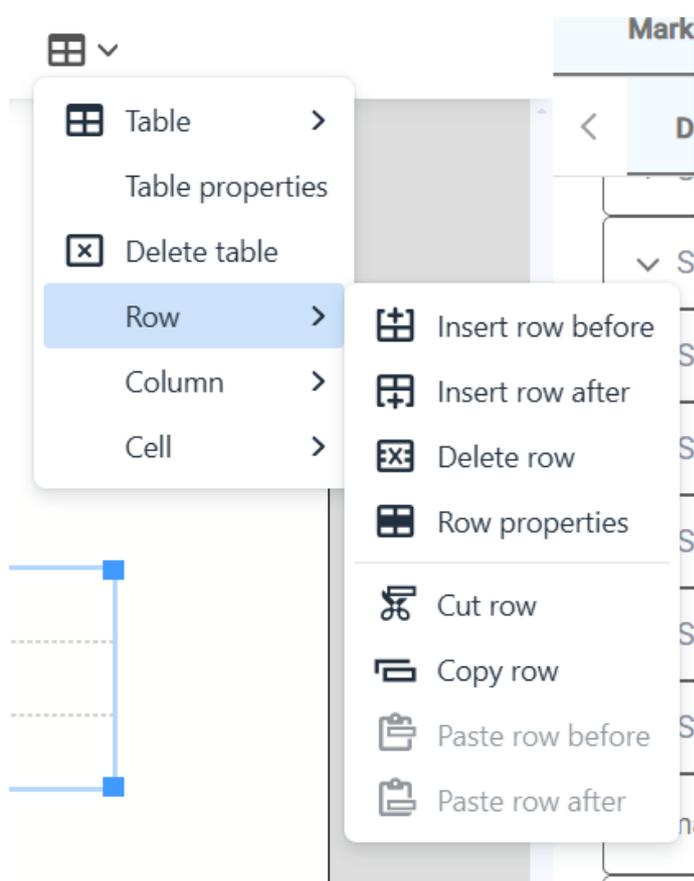


Figura 4.4: Inserimento di tabelle

L'inserimento di una nuova tabella avviene tramite il pulsante "Table" presente nella barra degli strumenti. Cliccando su questo pulsante si aprirà una griglia interattiva che permette di selezionare il numero desiderato di righe e colonne trascinando il cursore.

Una volta inserita la tabella, selezionandola si aprirà un menu contestuale che fornisce accesso rapido alle principali funzionalità di modifica.

## 4.2.2 Inserimento di righe e colonne



**Figura 4.5:** Menu per la gestione delle righe

L'editor offre diverse modalità per aggiungere righe e colonne a una tabella esistente. Per inserire una nuova colonna, è possibile utilizzare i pulsanti "Insert column before" o "Insert column after", accessibili sia dal menu contestuale che dal sotto-

menu "Column" del pulsante "Table". Questi comandi permettono rispettivamente di aggiungere una colonna prima o dopo quella selezionata.

Allo stesso modo, per aggiungere una nuova riga è possibile utilizzare i pulsanti "Insert row before" o "Insert row after", disponibili nel menu contestuale o nel sottomenu "Row" del pulsante "Table". Questi comandi consentono di inserire una riga rispettivamente sopra o sotto quella selezionata.

### 4.2.3 Rimozione di tabelle, righe e colonne



Figura 4.6: Rimozione della tabella attraverso il menu contestuale

Per rimuovere una tabella è possibile utilizzare il pulsante "Delete Table" presente sia nel menu contestuale che nel menu "Table".

Analogamente, per eliminare una riga o colonna specifica è possibile utilizzare i pulsanti "Delete Row" o "Delete Column", accessibili dal menu contestuale oppure rispettivamente dai sottomenu "Row" e "Column" del menu "Table".

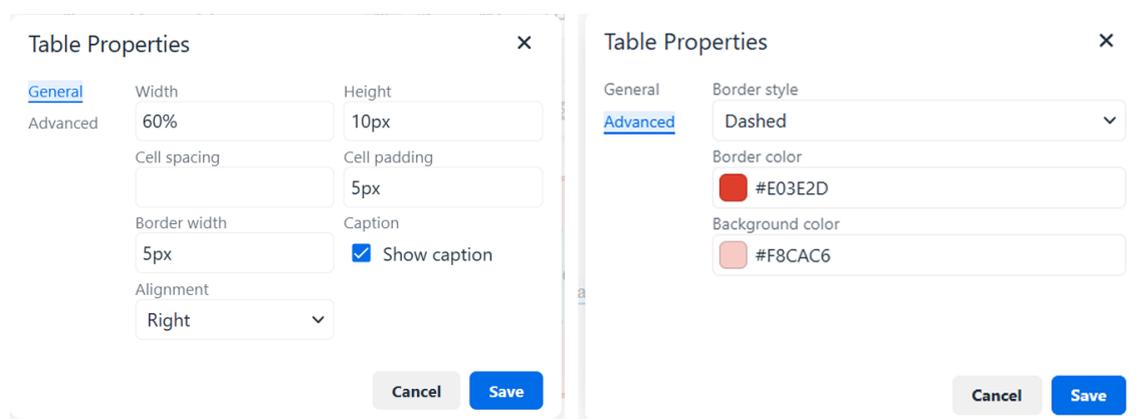
### 4.2.4 Taglia, copia e incolla righe e colonne

L'editor fornisce funzionalità per tagliare, copiare e incollare righe e colonne attraverso appositi pulsanti nei menu contestuali. Quando si copia o taglia una riga o colonna, vengono preservati sia il contenuto che lo stile di tutte le celle.

È possibile incollare gli elementi copiati sia prima che dopo la riga o colonna selezionata utilizzando rispettivamente i pulsanti "Paste Row/Column Before" e "Paste

Row/Column After”. Questo permette una gestione flessibile della struttura della tabella, consentendo di riorganizzare facilmente i dati secondo le proprie necessità.

## 4.2.5 Modifica delle proprietà delle tabelle



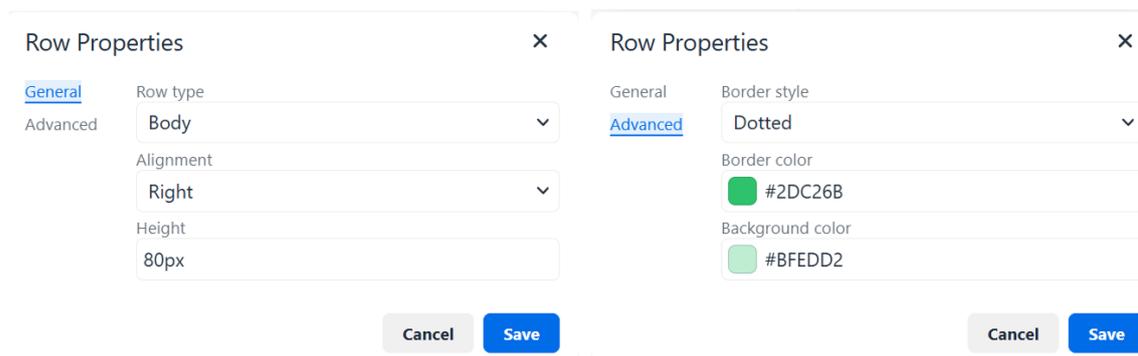
**Figura 4.7:** Menu delle proprietà delle tabelle

Il menu delle proprietà della tabella, accessibile tramite il pulsante "Table properties" nel menu contestuale o nel menu "Table", offre diverse opzioni per personalizzare l'aspetto della tabella selezionata.

È possibile definire le dimensioni della tabella modificandone larghezza e altezza, regolare la spaziatura interna delle celle attraverso i valori di spacing e padding, e aggiungere o rimuovere una didascalia descrittiva. L'allineamento della tabella all'interno del documento può essere controllato per ottenere il posizionamento desiderato.

Per quanto riguarda l'aspetto visivo, il menu permette di personalizzare completamente i bordi della tabella, modificandone spessore, stile e colore, nonché di impostare un colore di sfondo per l'intera struttura.

## 4.2.6 Modifica delle proprietà delle righe

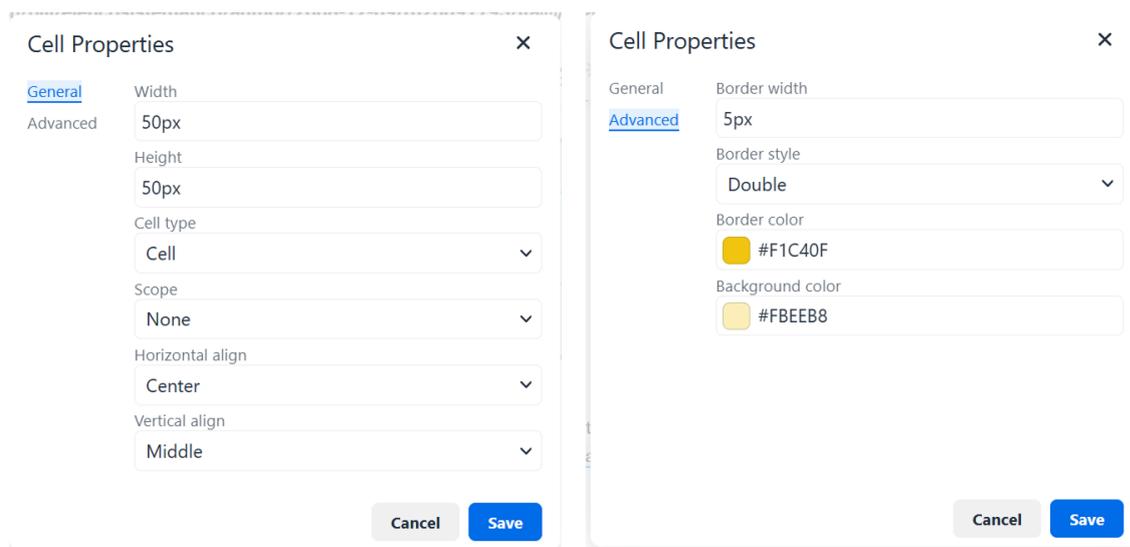


**Figura 4.8:** Menu delle proprietà delle righe

Il menu delle proprietà delle righe, accessibile tramite il pulsante "Table > Row > Row properties", offre diverse opzioni per personalizzare la riga selezionata.

Attraverso questo menu è possibile definire il tipo di riga, scegliendo tra body, header o footer, e controllare l'allineamento del contenuto all'interno delle celle. È inoltre possibile regolare l'altezza della riga e personalizzarne l'aspetto visivo modificando lo spessore, lo stile e il colore dei bordi, nonché impostare un colore di sfondo specifico.

## 4.2.7 Modifica delle proprietà delle celle



**Figura 4.9:** Menu delle proprietà delle celle

Il menu delle proprietà delle celle, accessibile tramite il pulsante "Table > Cell > Cell properties", permette di personalizzare la cella attualmente selezionata.

Attraverso questo menu è possibile definire le dimensioni della cella modificandone altezza e larghezza, specificare se si tratta di una cella di intestazione o una cella normale, e definirne lo scope che indica per quali righe o colonne la cella funge da intestazione.

Il menu consente inoltre di controllare l'allineamento del contenuto sia in orizzontale che in verticale. Per quanto riguarda l'aspetto visivo, è possibile personalizzare i bordi della cella modificandone spessore, stile e colore, nonché impostare un colore di sfondo specifico.

## 4.2.8 Salvataggio delle tabelle

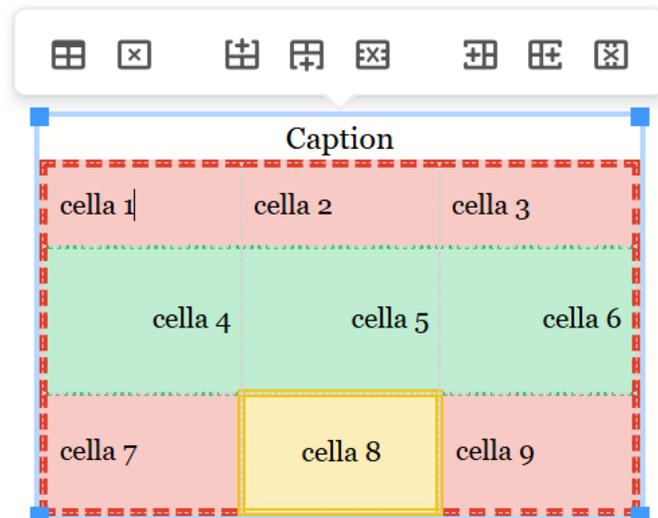


Figura 4.10: Esempio di tabella con proprietà modificate

Le tabelle, analogamente alle immagini, vengono salvate automaticamente nel momento in cui l'utente salva il documento attraverso il pulsante "File > Save". Questo garantisce che tutte le modifiche apportate alle proprietà e alla struttura delle tabelle vengano preservate correttamente.

## 4.3 Gestione dei package

L'introduzione del concetto di package in Simplex ha portato a una gestione più strutturata e standardizzata dei documenti. Questa nuova organizzazione permette di gestire in modo coerente tutti i componenti del documento, inclusi file XML, immagini e fogli di stile, seguendo le specifiche del package manifestation di Akoma Ntoso.

In precedenza, l'utente poteva creare solo singoli documenti XML, dovendo gestire manualmente attraverso il filesystem i riferimenti a componenti esterni come immagini e fogli di stile. Con la nuova implementazione, l'utente può concentrarsi esclusivamente sulla creazione del contenuto, poiché la gestione dei componenti

multimediali, dei fogli di stile e dei relativi riferimenti viene effettuata in modo completamente automatico dal sistema.

Ogni documento in Simplex è ora gestito come un package, esportabile come file zip con estensione ".akn". Il package contiene al suo interno il componente principale XML e tutti gli altri componenti, come immagini e fogli di stile, organizzati secondo le regole della naming convention di Akoma Ntoso.

Quando l'utente aggiunge contenuti multimediali o personalizza lo stile del documento, il sistema gestisce automaticamente l'organizzazione dei file all'interno del package. Le risorse multimediali vengono salvate nella cartella "media" con identificatori univoci, mentre i fogli di stile CSS vengono generati e salvati nella stessa cartella utilizzando il nome del documento XML associato.

# Capitolo 5

## Implementazione della gestione di immagini, tabelle e package

In questo capitolo verranno illustrate nel dettaglio le implementazioni delle funzionalità di gestione delle immagini, delle tabelle, dello stile e dei package. Verranno analizzati gli aspetti tecnici e le scelte implementative adottate per ciascun componente, descrivendo le interazioni con il resto del sistema.

### 5.1 Implementazione della gestione delle immagini

#### 5.1.1 Componenti per la gestione delle immagini

Le immagini nel documento vengono inserite nell'editor TinyMCE come elementi HTML "figure". L'elemento "figure" è un contenitore semantico utilizzato per racchiudere contenuti autonomi, in questo caso immagini (elemento "img"), spesso accompagnati da una didascalia (elemento "figcaption").

Il modulo "toolbar.ts" gestisce la configurazione della barra degli strumenti dell'editor TinyMCE. Questo modulo contiene la funzione "setupToolbarButtons", che utilizza l'API di TinyMCE per registrare nuovi pulsanti nella barra degli strumenti.

Attraverso questa funzione è stato aggiunto il pulsante "Add/Edit Image" che apre il dialog per la gestione delle immagini.

Per il dialog è stato creato un nuovo componente Angular denominato "ImageDialogComponent" che gestisce il caricamento di nuove immagini e la modifica di quelle esistenti. Tramite l'interfaccia è possibile impostare le dimensioni (attributi "width" e "height") con o senza blocco delle proporzioni, specificare un titolo (attributo "title") e aggiungere una didascalia opzionale (elemento "figcaption"). Il componente implementa la validazione dei formati supportati, che includono JPG, PNG, GIF e TIF.

Il comportamento del pulsante "Add/Edit Image" varia in base al contesto. Quando nessun elemento "figure" è selezionato, il dialog si apre in modalità inserimento, permettendo di caricare una nuova immagine e configurarne le proprietà. Alla conferma tramite il tasto "Save", viene creato un nuovo elemento "figure" nel documento, contenente un elemento "img" e, se specificato, un elemento "figcaption". Gli attributi vengono impostati secondo i valori inseriti dall'utente.

Nel caso in cui sia selezionato un elemento "figure", il dialog si apre in modalità modifica, mostrando i valori correnti delle proprietà dell'immagine. L'utente può modificare questi valori e l'immagine stessa. In questa modalità, il tasto "Save" aggiorna gli elementi esistenti con i nuovi valori, mentre un pulsante "Delete" aggiuntivo permette di rimuovere l'immagine dal documento.

Il componente gestisce anche il salvataggio temporaneo delle immagini a livello locale. Quando un'immagine viene caricata, questa viene convertita in un oggetto Blob e viene generato un URL temporaneo attraverso il metodo "createObjectURL" della web API "URL". L'URL generato viene assegnato all'attributo "src" dell'elemento "img" corrispondente nel documento. Viene inoltre aggiunto l'attributo "data-local" all'elemento, che servirà come marcatore per identificare le immagini che devono ancora essere salvate permanentemente sul server durante il successivo salvataggio del documento.

### 5.1.2 Salvataggio e recupero delle immagini dal server

Come per qualsiasi altra modifica apportata al documento, le immagini vengono salvate sul server solo quando l'utente utilizza il pulsante "File > Save". Nella configurazione standard di Simplex, definita nel file "config.json", questa azione attiva il metodo "saveDocument" del service "FileManagerService", che gestisce le operazioni del menu "File". La funzione recupera il contenuto HTML del documento e lo passa al metodo "saveDocument" di "aknEditorService.ts", il service principale dell'editor. Quest'ultimo utilizza gli operatori RxJS di Angular per gestire il flusso di dati asincroni ed effettuare le necessarie conversioni, che verranno descritte in dettaglio nelle sezioni 5.2 e 5.4.

Il documento viene quindi elaborato dal metodo "uploadImages". Questa funzione identifica le immagini marcate con l'attributo "data-local", che devono essere salvate sul server. Attraverso l'operatore "forkJoin" di RxJS, viene effettuato il caricamento parallelo di tutte le immagini trovate. Per ciascuna immagine viene invocato il metodo "saveImage" del "documentService", il service responsabile delle operazioni di input/output con il server. Il metodo converte l'immagine in formato base64 e ne richiede il caricamento nel database eXist attraverso l'API del server. L'immagine sarà poi salvata all'interno della cartella "media" del package con un id univoco come nome del file. Se la cartella non esiste, viene creata automaticamente.

Una volta completato il caricamento sul server, il metodo "uploadImages" rimuove l'attributo "data-local" dall'elemento "img" corrispondente, poiché questo attributo era utilizzato solo come marcatore temporaneo per identificare le immagini non ancora salvate sul server. In seguito aggiorna l'attributo "src" dell'elemento "img", sostituendo l'URL locale temporaneo con il path dell'immagine. Questo è composto dal path relativo della cartella "media" del package, seguito dall'id univoco assegnato all'immagine durante il salvataggio. Ad esempio per un'immagine con id "img\_1" il path sarà "media/img\_1".

"AknTinyMceIntegrator" è un service che funge da intermediario tra l'editor TinyMCE e gli altri servizi dell'applicazione. Quando viene aperto un documento, sia esso caricato dal server, appena creato o importato dall'utente, viene invocato il metodo "showDocument" di questo service.

All'interno di "showDocument" viene chiamata la funzione "fetchImages", responsabile del recupero delle immagini dal server. Il recupero avviene in modo parallelo per tutte le immagini presenti nel documento. Per ogni elemento "img", viene calcolato il path dell'immagine utilizzando l'id memorizzato nell'attributo "src". Questo path viene quindi passato al metodo "getImage" del "documentService", che si interfaccia con l'API del server per ottenere l'immagine.

Una volta recuperata, l'immagine viene convertita in formato Blob e viene generato un object URL che andrà a popolare l'attributo "src" dell'elemento "img". L'id originale dell'immagine viene preservato nell'attributo "data-src" per essere utilizzato nelle successive operazioni di conversione del documento.

## 5.2 Conversione da HTML ad Akoma Ntoso

Il "ConverterService" è il service responsabile della conversione dei documenti tra diversi formati. In particolare, i metodi "HTMLToAkn" e "AknToHTML" gestiscono rispettivamente la conversione da HTML a Akoma Ntoso e viceversa. La conversione viene realizzata attraverso l'utilizzo di XSLT (eXtensible Stylesheet Language Transformations), un linguaggio di trasformazione per documenti XML. Attraverso specifici fogli di stile XSLT vengono definite le regole di mappatura tra gli elementi HTML e i corrispondenti elementi Akoma Ntoso. Questo approccio garantisce che durante la conversione vengano rispettate tutte le specifiche dello standard Akoma Ntoso, mantenendo la corretta struttura del documento.

Prima del salvataggio di un documento, l'AknEditorService invoca la funzione "HTMLToAkn" per trasformare il documento dal formato HTML, necessario per la visualizzazione e modifica nell'editor TinyMCE, al formato Akoma Ntoso. Viceversa, quando un documento Akoma Ntoso viene aperto, creato o importato nell'editor, il metodo "AknToHTML" viene chiamato per convertire il documento da Akoma Ntoso ad HTML.

### 5.2.1 Conversione delle immagini

La conversione degli elementi "figure" e "figcaption", utilizzati nell'editor TinyMCE, richiede particolare attenzione poiché questi non sono presenti nello schema di Akoma Ntoso. Per gestire questa differenza, sono state definite specifiche regole di trasformazione nel foglio di stile "HTMLToAkn.xsl". Durante la conversione da HTML ad Akoma Ntoso, gli elementi "figure" vengono trasformati in elementi "div" con la classe "figure", mentre gli elementi "figcaption" diventano elementi "div" con la classe "figcaption". Il contenuto di entrambi gli elementi viene opportunamente racchiuso in elementi "p".

Il processo inverso avviene nella conversione da Akoma Ntoso ad HTML: il foglio di stile "AknToHTML.xsl" si occupa di convertire gli elementi "div" con le rispettive classi nei corrispondenti elementi HTML "figure" e "figcaption", permettendo così la corretta visualizzazione e modifica nell'editor.

### 5.2.2 Conversione delle tabelle

La conversione delle tabelle tra HTML e Akoma Ntoso è relativamente semplice poiché entrambi i formati utilizzano elementi simili per rappresentare le tabelle. I tag per la definizione delle tabelle come "table", "tr" e "td" vengono mantenuti in quanto definiti sia nello schema di HTML che in quello di Akoma Ntoso.

## 5.3 Implementazione della gestione delle tabelle

Il secondo pulsante aggiunto al modulo "toolbar.ts" è il pulsante "Table". Attraverso le funzioni di base di TinyMCE, è stato creato un componente "menuItem" personalizzato alle esigenze dell'editor Simplex. Questo menu permette di creare, modificare e rimuovere tabelle, righe e colonne. I "menuItem" e "submenuItem" di "Table" possono essere riassunti così:

- **insertTable**: permette di inserire una nuova tabella attraverso una griglia interattiva.
- **tableProps**: permette di modificare le proprietà della tabella selezionata.

```
<figure pattern="figure" name="figure" id="figure_1">
  
  <figcaption
    pattern="figcaption"
    name="figcaption"
    id="figure_1__figcaption_1"
  >
    example
  </figcaption>
</figure>
```

**Figura 5.1:** Esempio di elemento HTML figure utilizzato nell'editor

- **deleteTable:** permette di rimuovere la tabella selezionata.
- **Row:** Sottomenu contenente le funzioni per gestire le righe della tabella.
- **Column:** Sottomenu contenente le funzioni per gestire le colonne della tabella.
- **Cell:** Sottomenu contenente le funzioni per gestire le celle della tabella.

Il componente "table\_toolbar" permette inoltre di gestire le tabelle attraverso un menu contestuale che viene mostrato alla selezione di una tabella. Questo permette di effettuare modifiche rapide alla struttura della tabella, senza dover utilizzare il menu principale. Questa toolbar contiene i seguenti pulsanti:

- **tableProps:** permette di modificare le proprietà della tabella selezionata.
- **tableDelete:** permette di rimuovere la tabella selezionata.

```
<div class="figure" eid="figure_1">
  <p>
    
  </p>
  <div class="figcaption" eid="figure_1__figcaption_1">
    <p>example</p>
  </div>
</div>
```

Figura 5.2: Esempio di elemento figure dopo la conversione in Akoma Ntoso

- **tableInsertRowBefore**: permette di inserire una nuova riga prima della riga selezionata.
- **tableInsertRowAfter**: permette di inserire una nuova riga dopo la riga selezionata.
- **tableDeleteRow**: permette di rimuovere la riga selezionata.
- **tableInsertColBefore**: permette di inserire una nuova colonna prima della colonna selezionata.
- **tableInsertColAfter**: permette di inserire una nuova colonna dopo la colonna selezionata.
- **tableDeleteCol**: permette di rimuovere la colonna selezionata.

Per garantire la conformità della gestione delle tabelle con le regole di Akoma Ntoso sono state necessarie diverse modifiche. Queste permettono il corretto funzionamento anche in presenza di situazioni complesse come tabelle annidate o l’inserimento di immagini all’interno delle celle delle tabelle, assicurando che la struttura gerarchica del documento rimanga sempre conforme allo standard Akoma Ntoso. Le principali modifiche implementate per supportare questi casi d’uso vengono illustrate in dettaglio nei paragrafi seguenti.

```

<table eid="table_1">
  <tr eid="table_1__tr_1">
    <td eid="table_1__tr_1__td_1">
      <p eid="table_1__tr_1__td_1__p_1">Cella 1</p>
    </td>
    <td eid="table_1__tr_1__td_2">
      <p eid="table_1__tr_1__td_2__p_1">Cella 2</p>
    </td>
  </tr>
  <tr eid="table_1__tr_2">
    <td eid="table_1__tr_2__td_1">
      <p eid="table_1__tr_2__td_1__p_1">Cella 3</p>
    </td>
    <td eid="table_1__tr_2__td_2">
      <p eid="table_1__tr_2__td_2__p_1">Cella 4</p>
    </td>
  </tr>
</table>

```

**Figura 5.3:** Esempio di tabella in Akoma Ntoso

L'inserimento di nuove tabelle nell'editor avviene attraverso l'evento "mceInsertContent", che viene intercettato dal componente "AknTinyMceIntegrator". Quest'ultimo funge da intermediario tra l'editor e i servizi dell'applicazione, e gestisce l'evento richiamando la funzione "paste" del modulo "keyboard-actions.ts".

L'evento "mceInsertContent" viene utilizzato dall'editor TinyMCE sia per l'inserimento di nuove tabelle che quando si incolla del contenuto. Per garantire la conformità con le regole dell'editor, è necessario identificare quando viene inserita una tabella e apportare le modifiche appropriate. Queste includono ad esempio l'aggiunta degli attributi "pattern" e "name". Inoltre, il contenuto di ogni cella deve essere racchiuso in un elemento generico di pattern "block", assicurando così il corretto funzionamento delle regole di inserimento e di gerarchia degli elementi.

Quando una tabella viene modificata attraverso l'inserimento di nuove righe o colonne, viene generato l'evento "TableModified", che viene gestito dall'AknTinyMceIntegrator. Anche in questo caso, il componente applica le stesse regole di conformità utilizzate durante la creazione iniziale della tabella: assegna gli attributi necessari e garantisce la corretta strutturazione delle nuove righe, colonne e celle aggiunte. Questo processo assicura che anche le parti modificate della tabella rispettino le regole dell'editor.

## 5.4 Gestione dello stile

Lo standard Akoma Ntoso permette di applicare l'attributo "style" a qualsiasi elemento presente nella sezione di contenuto del documento. Tuttavia, per migliorare la leggibilità e semplificare la manutenzione degli stili, si è optato per una separazione tra il documento e il foglio CSS degli stili applicati dall'utente.

L'utente può modificare lo stile del documento direttamente dall'editor, senza doversi preoccupare di scrivere codice CSS, la cui generazione verrà gestita dall'applicazione.

Quando l'utente modifica l'aspetto del documento attraverso l'editor TinyMCE, gli stili vengono inizialmente applicati come attributi inline agli elementi HTML. Al momento del salvataggio, gli stili vengono estratti e organizzati in un file CSS dedicato, che viene salvato nella cartella "media" del package con lo stesso nome del documento XML ma con estensione ".css".

Per gestire questa separazione, il ConverterService è stato arricchito con il nuovo metodo "HTMLToAKNAndCSS", che si occupa di convertire il documento HTML con stili inline in due componenti distinte: un documento Akoma Ntoso (privo di stili inline) e il file CSS associato. Il documentService è stato analogamente esteso con il metodo "saveFileAndCSS", che gestisce il salvataggio di entrambi i file attraverso l'API del server.

Durante il processo di salvataggio, la funzione "saveDocument" dell'AknEditorService orchestrerà prima la conversione e poi il salvataggio sia del documento che del file CSS, quest'ultimo solo se sono presenti stili aggiunti dall'utente. In fase di apertura

del documento, il metodo "showDocument" dell'AknTinyMceIntegrator recupererà il file CSS associato (se esistente) attraverso la funzione "fetchStyles" e applicherà gli stili come attributi inline, garantendo così la compatibilità con TinyMCE.

## 5.5 Gestione dei package

L'implementazione della gestione dei package ha richiesto una completa ristrutturazione dell'organizzazione dei documenti di Simplex. Il lavoro si è concentrato su due aspetti principali: la riorganizzazione della struttura dei documenti secondo le specifiche Akoma Ntoso e l'automazione della gestione dei componenti multimediali all'interno di questa nuova struttura.

La struttura precedente, basata su un sistema di cartelle e file XML annidati, è stata trasformata in un'organizzazione conforme alle specifiche del package manifestation di Akoma Ntoso. Nella nuova implementazione, ogni documento è rappresentato da un package, contenente il componente principale a livello manifestation (file XML) e tutti i componenti a cui fa riferimento, organizzati secondo le regole definite dallo standard.

Secondo le naming convention di Akoma Ntoso, un package manifestation è un file zip con estensione ".akn" che contiene uno o più file memorizzati allo stesso livello. Il componente principale del documento viene salvato come "main.xml", mentre i componenti multimediali vengono organizzati all'interno di una cartella dedicata denominata "media".

I metodi del DocumentService sono stati aggiornati per implementare questa struttura. Durante il salvataggio, il metodo "saveImage" colloca le immagini nella cartella "media", assegnando loro un ID univoco e l'estensione appropriata. Se necessario, la cartella "media" viene creata automaticamente. All'apertura di un documento, il metodo "getImage" recupera le immagini utilizzando percorsi relativi al package.

In modo analogo, per i fogli di stile, il metodo "saveFileAndCSS" salva il file CSS nella cartella "media", utilizzando lo stesso nome del documento XML ma con estensione ".css". All'apertura del documento, il metodo "getCSS" recupera il file di stile dalla cartella "media" quando presente.

La nuova implementazione della gestione dei package offre numerosi vantaggi rispetto alla struttura precedente. Innanzitutto, garantisce la piena conformità allo standard Akoma Ntoso, facilitando l'interoperabilità con altri eventuali sistemi che utilizzano lo stesso standard. Inoltre, semplifica notevolmente l'esperienza dell'utente, che può ora concentrarsi esclusivamente sulla creazione del contenuto, mentre la gestione dei componenti multimediali, dei fogli di stile e dei relativi riferimenti viene effettuata in modo completamente automatico dal sistema. Infine, la nuova struttura migliora la portabilità dei documenti, che possono essere facilmente esportati, condivisi e importati mantenendo intatta tutta la loro struttura e tutti i loro componenti.



# Capitolo 6

## Valutazione

Le funzionalità implementate nell'editor Simplex sono state sottoposte a un processo di valutazione attraverso test sistematici, condotti sia durante lo sviluppo che nella fase finale. Questo processo iterativo ha permesso di adattare e perfezionare le implementazioni in base ai feedback e ai requisiti emersi, con l'obiettivo di garantire l'efficacia, l'usabilità e la piena conformità alle specifiche di Akoma Ntoso. La metodologia di valutazione ha incluso test funzionali per verificare il corretto funzionamento delle implementazioni e test di validazione della conformità allo standard Akoma Ntoso.

### 6.1 Test funzionali

I test funzionali hanno verificato il corretto funzionamento delle nuove funzionalità implementate, analizzando in dettaglio ciascun componente sviluppato.

#### 6.1.1 Gestione delle immagini

Per quanto riguarda la gestione delle immagini, i test hanno verificato l'inserimento di immagini nei formati supportati (JPG, PNG, GIF, TIF), il ridimensionamento con e senza vincolo delle proporzioni, l'aggiunta e modifica di didascalie, il salvataggio e recupero delle immagini dal server, e la corretta conversione tra HTML e Akoma

Ntoso. I risultati hanno confermato il corretto funzionamento di tutte le operazioni, con particolare attenzione alla gestione degli attributi.

### **6.1.2 Gestione delle tabelle**

Nel caso delle tabelle, sono state testate la creazione con diverse dimensioni, l'inserimento e rimozione di righe e colonne, le operazioni di copia, taglia e incolla, la modifica delle proprietà di tabelle, righe e celle, la gestione di tabelle annidate e l'inserimento di contenuti complessi nelle celle. I test hanno evidenziato la robustezza dell'implementazione anche in scenari complessi, come la gestione di tabelle annidate o l'inserimento di immagini nelle celle.

### **6.1.3 Gestione dei package**

Per i package, la valutazione ha riguardato la creazione e il salvataggio e l'organizzazione automatica dei componenti, la gestione della cartella media sia per le immagini che per i fogli di stile. I test hanno confermato la corretta implementazione della struttura dei package secondo le specifiche di Akoma Ntoso.

## **6.2 Test di validazione**

La conformità allo standard Akoma Ntoso è stata verificata attraverso la validazione dei documenti generati, la verifica della corretta struttura dei package e il controllo della corretta conversione tra HTML e AKN. I test hanno confermato la piena conformità delle implementazioni alle specifiche dello standard.

## **6.3 Risultati della valutazione**

La valutazione complessiva delle nuove funzionalità implementate in Simplex è decisamente positiva. I test hanno dimostrato il corretto funzionamento di tutte le implementazioni, un'elevata usabilità con significativi miglioramenti rispetto a soluzioni precedenti e una piena conformità allo standard Akoma Ntoso.

<b>Funzionalità</b>	<b>Versione precedente</b>	<b>Versione attuale</b>
<b>Gestione delle immagini</b>		
Inserimento di immagini	No	Sì
Modifica delle immagini	No	Sì
Ridimensionamento delle immagini	No	Sì
Rimozione delle immagini	No	Sì
Salvataggio locale delle immagini	No	Sì
<b>Gestione delle tabelle</b>		
Inserimento di tabelle	No	Sì
Inserimento di righe e colonne	No	Sì
Rimozione di tabelle, righe e colonne	No	Sì
Taglia, copia e incolla righe e colonne	No	Sì
Modifica delle proprietà delle tabelle	No	Sì
Modifica delle proprietà delle righe	No	Sì
Modifica delle proprietà delle celle	No	Sì
Salvataggio locale delle tabelle	No	Sì
<b>Gestione dei package</b>		
Implementazione del concetto di package manifestation	No	Sì
Gestione automatica dei componenti multimediali	No	Sì
Salvataggio e recupero dei componenti multimediali dal database	No	Sì
<b>Gestione dei fogli di stile</b>		
Generazione dei fogli di stile	No	Sì
Gestione automatica dei fogli di stile	No	Sì
Salvataggio e recupero di fogli di stile dal database	No	Sì
<b>Conversione HTML-AKN</b>		
Conversione delle immagini da HTML a AKN e viceversa	No	Sì
Conversione delle tabelle da HTML a AKN e viceversa	No	Sì

**Tabella 6.1:** Confronto delle funzionalità tra la versione precedente e la versione attuale di Simplex



# Capitolo 7

## Conclusioni

Il lavoro di tesi ha portato all'implementazione di tre componenti chiave nell'editor Simplex: la gestione delle immagini, la gestione delle tabelle e l'implementazione del concetto di package manifestation. Queste funzionalità hanno contribuito a migliorare significativamente le capacità dell'editor nella creazione e modifica di documenti legali strutturati.

La gestione delle immagini permette ora agli utenti di inserire, modificare e organizzare contenuti grafici in modo intuitivo, mantenendo la piena conformità con lo standard Akoma Ntoso. L'implementazione include funzionalità complete per il ridimensionamento, l'aggiunta di didascalie e la gestione dello stile, offrendo un'esperienza d'uso paragonabile a quella di editor di documenti tradizionali.

La gestione delle tabelle fornisce un insieme di strumenti per la creazione e modifica di tabelle strutturate. Le funzionalità implementate includono l'inserimento e la rimozione di righe e colonne, la personalizzazione dello stile e il supporto per celle di intestazione, permettendo la creazione di tabelle complesse mantenendo la conformità con lo schema Akoma Ntoso.

L'implementazione del concetto di package manifestation ha portato a una gestione più strutturata e standardizzata dei documenti. Questa nuova organizzazione permette di gestire in modo coerente tutti i componenti del documento, inclusi file XML, immagini e fogli di stile, seguendo le specifiche di Akoma Ntoso.

La gestione dello stile è stata implementata seguendo il principio di separazione tra contenuto e presentazione, con la generazione automatica di fogli di stile CSS. Questo approccio migliora la manutenibilità dei documenti e facilita la gestione coerente degli stili attraverso l'interfaccia utente.

Particolare attenzione è stata dedicata alla conversione tra formati HTML e Akoma Ntoso, garantendo la corretta rappresentazione degli elementi nel documento finale. L'approccio adottato, basato su trasformazioni XSLT, assicura che le conversioni mantengano la struttura semantica del documento rispettando le regole dello standard.

## 7.1 Sviluppi futuri

Il lavoro svolto apre la strada a diversi possibili sviluppi futuri:

- **Ampliamento delle funzionalità di gestione delle immagini:** Il componente "ImageDialogComponent" può essere facilmente esteso per supportare ulteriori funzionalità di editing delle immagini, come la personalizzazione del bordo dell'immagine. Inoltre, si potrebbe ampliare il supporto a formati di immagine addizionali oltre a quelli già gestiti.
- **Supporto per contenuti multimediali:** L'implementazione attuale potrebbe essere estesa per gestire altri tipi di contenuti multimediali. Questi verrebbero salvati nella cartella "media" del package, mantenendo la stessa organizzazione utilizzata per le immagini.
- **Integrazione di PDF:** L'editor potrebbe essere arricchito con funzionalità per incorporare e visualizzare documenti PDF esterni, permettendo di referenziarli e mostrarli all'interno del documento principale.
- **Gestione avanzata dei package:** Il sistema di package potrebbe essere potenziato con funzionalità aggiuntive per la modifica e visualizzazione dei package.
- **Estensione delle funzionalità di stile:** La gestione degli stili potrebbe essere arricchita con strumenti per la personalizzazione e la formattazione

degli altri elementi del documento.



# Riferimenti bibliografici

- [CPCdB] Luca Cervone, Monica Palmirani, CIRSfid, and Università di Bologna. Lime, the Language Independent Markup Editor. <http://lime.cirsfid.unibo.it>. Ultima visita: Febbraio 2025.
- [CPV] Luca Cervone, Monica Palmirani, and Fabio Vitali. Akoma Ntoso - XML for parliamentary, legislative and judiciary documents. <http://akomantoso.info/>. Ultima visita: Febbraio 2025.
- [Dzi22] Marek Patryk Dziejczak. Redesign and implementation of an AkomaNtoso based web editor for the markup of legal documents. Tesi di laurea triennale Cds Informatica, Università di Bologna, 2022. <https://amslaurea.unibo.it/25313>.
- [Fer24] Giulia Ferrigno. ConsolidationViewer: uno strumento web per il consolidamento di documenti legislativi. Tesi di laurea triennale Cds Informatica, Università di Bologna, 2024. <https://amslaurea.unibo.it/34175/>.
- [Pog24] Mattia Poggioli. Gestione di eventi utente in un editor di documenti legali basato su Akoma Ntoso. Tesi di laurea triennale Cds Informatica, Università di Bologna, 2024. <https://amslaurea.unibo.it/30983/>.
- [VPa] Fabio Vitali and Monica Palmirani. Akoma Ntoso: principi generali. [https://www.agid.gov.it/sites/default/files/repository\\_files/documentazione/20150525\\_akoma\\_architettura\\_unibo\\_f\\_vitali\\_v\\_1\\_0.pdf](https://www.agid.gov.it/sites/default/files/repository_files/documentazione/20150525_akoma_architettura_unibo_f_vitali_v_1_0.pdf). Ultima visita: Febbraio 2025.
- [VPb] Fabio Vitali and Monica Palmirani. La Naming Convention di Akoma Ntoso. [https://www.agid.gov.it/sites/default/files/repository\\_](https://www.agid.gov.it/sites/default/files/repository_)

files/documentazione/20150525\_akoma\_naming\_convention\_unibo\_f\_vitali\_v\_1\_0.pdf. Ultima visita: Febbraio 2025.

- [VP18a] Fabio Vitali and Monica Palmirani. Akoma Ntoso Version 1.0. <https://www.oasis-open.org/standard/akn-v1-0/>, August 2018.
- [VP18b] Fabio Vitali and Monica Palmirani. Akoma Ntoso Version 1.0. Part 1: Xml Vocabulary. <https://docs.oasis-open.org/legaldocml/akn-core/v1.0/akn-core-v1.0-part1-vocabulary.html>, August 2018.
- [VP19] Fabio Vitali and Monica Palmirani. Akoma Ntoso Naming Convention Version 1.0. <https://docs.oasis-open.org/legaldocml/akn-nc/v1.0/akn-nc-v1.0.html>, February 2019.
- [Zec22] Andrea Zecca. Sviluppo di un editor di metadati per Simplex. Tesi di laurea triennale Cds Informatica, Università di Bologna, 2022. <https://amslaurea.unibo.it/26240/>.