

Dokumentacja techniczna
schematów XML Schema
opisujących strukturę projektów aktów prawnych
w systemie Konsultacje On-line

Wersja 1.0

Warszawa, 20 września 2012 r.

Spis treści

1	Wstęp.....	4
2	Schematy, dokumenty schematów i przestrzenie nazw	4
2.1	Elementy główne	5
2.2	Przestrzenie nazw	5
2.3	Budowa fizyczna schematów	5
3	Zastosowane wzorce, rozwiązania i konwencje.....	6
3.1	Nazewnictwo	6
3.2	Identyfikatory jednostek systematyzacyjnych i redakcyjnych	6
3.2.1	Wykorzystanie identyfikatorów przy eksporcie i imporcie dokumentów	6
3.2.2	Postać identyfikatorów	7
3.3	Elementy globalne i lokalne	7
3.4	Typy nazwane.....	7
3.5	Grupy.....	8
4	Budowa logiczna aktów prawnych	8
4.1	Nagłówek.....	8
4.1.1	Ustawa.....	8
4.1.2	Rozporządzenie	9
4.1.3	Inny dokument.....	10
4.2	Jednostki systematyzacyjne	10
4.3	Jednostki redakcyjne.....	11
4.3.1	Wyliczenia.....	12
4.3.2	Numery jednostek redakcyjnych.....	15
4.4	Akapity	15
4.4.1	Tabele	15
4.4.2	Grafika rastrowa.....	16
4.4.3	Grafika wektorowa.....	17
4.4.4	Wzory matematyczne	17
4.5	Zawartość tekstowa	18
4.5.1	Formatowanie.....	19
4.5.2	Przypisy i odnośniki do przypisów	19
4.5.3	Cytaty	20
4.5.4	Instrukcje nowelizacyjne.....	21
4.6	Załączniki	21
5	Znaczniki nowelizacyjne.....	21
5.1	Wskazanie nowelizowanego aktu prawnego.....	22
5.2	Dodanie jednostek redakcyjnych.....	23
5.3	Dodanie fragmentu przepisu.....	23
5.4	Dodanie fragmentu przepisu i jednostek redakcyjnych.....	25

5.5	Uchylenie jednostek redakcyjnych	25
5.6	Uchylenie fragmentu przepisu	26
5.7	Uchylenie aktu prawnego	26
5.8	Zastąpienie jednostek redakcyjnych	27
5.9	Zastąpienie fragmentu przepisu	27
5.10	Wprowadzenie ustępu	28
5.11	Likwidacja zbędnego ustępu	29
5.12	Ręczne zmiany fragmentu przepisu	29
5.13	Zmiany niewidoczne w treści aktu nowelizującego	29
6	Możliwość rozbudowy schematów	30
6.1	Przedefiniowywanie typów	30
6.2	Przedefiniowywanie grup	31
7	Wykorzystanie podpisu elektronicznego	31
7.1	Wyłączanie znaczników nowelizacyjnych spod podpisu elektronicznego	31
7.2	Podpisywanie zewnętrznych plików binarnych	32
8	Porównanie z innymi rozwiązaniami	32
8.1	EDAP	32
8.2	CEN MetaLex	34
8.2.1	Możliwość konwersji do formatu CEN MetaLex	35
8.3	Akoma Ntoso	36
8.4	Norme In Rete	37

1 Wstęp

Niniejszy dokument zawiera dokumentację techniczną schematów XML Schema opisujących strukturę projektów aktów prawnych. Schematy te będą wykorzystywane we wdrażanym przez Ministerstwo Gospodarki systemie Konsultacje On-Line, wspierającym proces tworzenia regulacji prawnych.

Dodatkiem do niniejszej dokumentacji są opisy poszczególnych elementów, atrybutów, typów danych i grup, umieszczone bezpośrednio w dokumentach schematów, w elementach *documentation*.

Wszystkie dokumenty schematów są zgodne z rekomendacją XML Schema 1.0 Second Edition z dnia 28 października 2004 r.¹

Grafiki prezentujące fragmenty schematów XML Schema zostały wygenerowane w programie XML Spy firmy Altova.

2 Schematy, dokumenty schematów i przestrzeń nazw

Niniejszy dokument opisuje trzy schematy definiujące strukturę różnych rodzajów aktów prawnych. Każdy z tych schematów deklaruje elementy i atrybuty w innej przestrzeni nazw (tj. ma inną docelową przestrzeń nazw). W poniższej tabeli są przedstawione wszystkie schematy wraz z nazwami przestrzeni nazw:

Schemat	Dokument schematu	Przeźstrzeń nazw
Nowe ustawy i rozporządzenia – <i>strict</i>	<code>strict.xsd</code>	http://www.mg.gov.pl/prawo/strict
Archiwalne ustawy i rozporządzenia – <i>loose</i>	<code>loose.xsd</code>	http://www.mg.gov.pl/prawo/loose
Inne dokumenty (uchwały, oświadczenia, itp.) – <i>base</i>	<code>base.xsd</code>	http://www.mg.gov.pl/prawo/base

Schemat *strict* charakteryzuje się najbardziej restrykcyjną kontrolą hierarchicznej struktury aktu prawnego. Poza nielicznymi, opisanymi w tym dokumencie wyjątkami, nie są w nim dopuszczalne odstępstwa od zasad techniki prawodawczej.

Schemat *loose* jest nieco mniej restrykcyjny i dopuszcza odstępstwa od zasad techniki prawodawczej, polegające na możliwości dowolnego zagnieżdżania jednostek redakcyjnych reprezentujących wyliczenia (punktów, liter, tiret i podwójnych tiret). Zgodnie z tym schematem, każda z wymienionych jednostek redakcyjnych może zawierać zagnieżdżoną każdą inną (np. tiret może zawierać zagnieżdżone punkty).

Schemat *base* ma najbardziej swobodną strukturę, pozwalającą na dowolne zagnieżdżanie w sobie jednostek redakcyjnych dowolnego rodzaju.

¹ Zob. <http://www.w3.org/standards/techs/xmlschema>. Od kwietnia 2012 r. jest dostępna wersja 1.1 specyfikacji XML Schema, jednak nie ma ona jeszcze charakteru standardu (w szczególności nie jest wspierana przez wiele narzędzi).

2.1 Elementy główne

W schemacie *strict* dostępne są dwa elementy główne: **ustawa** i **rozporządzenie**. Budowa ustawy i rozporządzenia jest niemal identyczna – różnią się one zaledwie kilkoma elementami, dlatego oba typy dokumentów są umieszczone w jednym schemacie i w jednej przestrzeni nazw oraz korzystają w większości z tych samych struktur.

Analogicznie, dwa elementy główne (**ustawa** i **rozporządzenie**) posiada schemat *loose*.

Schemat *base* ma jeden element główny – **dokument**, przy pomocy którego reprezentowane są wszystkie dokumenty inne niż ustawa lub rozporządzenie.

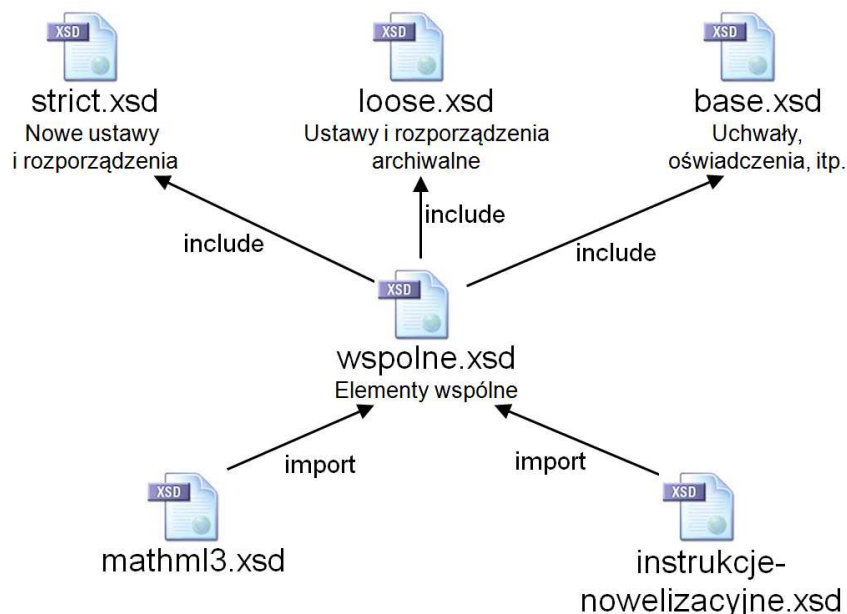
2.2 Przestrzeń nazw

Każdy z trzech schematów definiuje odrębną przestrzeń nazw. Takie rozwiązanie wynika z tego, że w każdym z trzech schematów występują elementy o identycznych nazwach (np. **punkt**, **litera**, **tiret**), ale model zawartości tych elementów jest w każdym schemacie inny. Należy zatem odróżnić od siebie semantycznie i składniowo elementy o tej samej nazwie, wykorzystywane w różnych kontekstach (jeden w aktach nowych, drugi w aktach archiwalnych, trzeci w innych dokumentach). Naturalnym mechanizmem pozwalającym na takie rozróżnienie są właśnie przestrzenie nazw.

Oddzielenie od siebie trzech stopni kontroli struktury jest szczególnie istotne ze względu na możliwość cytowania fragmentów aktów prawnych. Dzięki odrębnym przestrzeniom nazw możemy np. w akcie nowym cytować fragment aktu archiwalnego bez obawy o konflikt nazw.

2.3 Budowa fizyczna schematów

Na poniższym rysunku jest przedstawiona fizyczna budowa schematów.



Wszystkie trzy schematy korzystają ze wspólnych definicji i deklaracji podstawowych struktur, umieszczonych w dokumencie schematu `wspolne.xsd`. Dokument ten jest dołączany do schematów przy pomocy instrukcji `include`, co powoduje, że zawarte w nim deklaracje i definicje „wpadają” do przestrzeni nazw każdego ze schematów.

Dokument schematu `wspolne.xsd` importuje dwa kolejne dokumenty schematów:

- `instrukcje-nowelizacyjne.xsd` – elementy i atrybuty stanowiące instrukcje nowelizacyjne, umieszczone w odrębnej przestrzeni nazw <http://www.mg.gov.pl/prawo/nowelizacje> (zob. rozdział 5),
- `mathml3.xsd` – publicznie dostępny schemat języka MathML służącego do zapisywania wyrażeń matematycznych.

3 Zastosowane wzorce, rozwiązania i konwencje

3.1 Nazewnictwo

Nazwy definiowanych i deklarowanych konstrukcji (elementów, atrybutów, typów danych i grup) są sformułowane w języku polskim. Wyjątkiem od tej zasady są nazwy elementów i atrybutów mające identyczne znaczenie, co odpowiednie konstrukcje języka HTML (np. `b` – wytłuszczenie, `sup` – indeks górny, `href` – lokalizacja zewnętrznego pliku, elementy i atrybuty tabel).

Nazwy elementów, atrybutów i grup są pisane małymi literami. Jeżeli nazwa składa się z więcej niż jednego wyrazu, to jako separator wyrazów jest używany znak minus (np. `załącznik-binarny`).

Nazwy typów danych są pisane w konwencji programistycznej, tzn. zaczynają się z wielkiej litery, a kolejne wyrazy nazwy są rozpoczynane wielką literą. Dla odróżnienia od nazw elementów i atrybutów, nazwy typów są zakończone sufiksem `Typ` (np. `ZawartoscTekstowaTyp`).

Nazwy nie zawierają tzw. polskich znaków. Użycie znaków narodowych jest wprawdzie dopuszczalne w schematach, ale ze względu na ewentualne ograniczenia innych technologii (np. baz danych) polskie znaki narodowe są w nazwach zastąpione odpowiednikami łacińskimi.

3.2 Identyfikatory jednostek systematyzacyjnych i redakcyjnych

Każda jednostka systematyzacyjna i redakcyjna oraz każdy element, który może być powoływany w aktach nowelizujących, ma opcjonalny atrybut `id` – wewnętrzny identyfikator jednostki. Może on być wykorzystany do identyfikacji jednostek w systemie Konsultacje On-line. Sposób jego wykorzystania leży w gestii wykonawcy tego systemu. Poniższe wskazówki mają charakter fakultatywnych rekomendacji i mogą, lecz nie muszą być wykorzystane przez wykonawcę oprogramowania.

3.2.1 Wykorzystanie identyfikatorów przy eksporcie i imporcie dokumentów

Jeśli dokumenty będą eksportowane z systemu w celu ich edycji poza systemem, można umieszczać identyfikatory w eksportowanym dokumencie XML. Jeśli przyjmujemy, że w procesie zewnętrznej edycji dokumentu istniejące identyfikatory nie zostaną zmienione, to podczas importu zmodyfikowanego dokumentu można wykorzystać zasadę:

- jeśli jednostka redakcyjna w dokumencie ma identyfikator, to przy jego pomocy odnajdujemy odpowiedni obiekt reprezentujący tę jednostkę w systemie i aktualizujemy go,

- jeśli jednostka redakcyjna w dokumencie nie ma identyfikatora, wówczas uznajemy, że jest to nowa jednostka dodana w procesie edycji.

3.2.2 Postać identyfikatorów

Postać identyfikatorów może być dowolna – muszą one jedynie być unikatowe w ramach dokumentu. Warto rozważyć następujące dwa sposoby zapisu identyfikatorów:

- identyfikatory nieznaczące, np. `i15`, `i1243`
 - o zaleta: nie zależą od pozycji jednostki redakcyjnej w strukturze dokumentu oraz od numeru jednostki i jednostek nadrzędnych;
 - o wada: interpretacja identyfikatora (tj. określenie, którą jednostkę wskazuje) wymaga przejrzenia dokumentu, którego dotyczy identyfikator;
- identyfikatory znaczące, oparte na strukturze dokumentu, np. `art17a.ust3.pkt6`
 - o zaleta: identyfikatory czytelne także dla człowieka (mogą być wręcz tworzone ręcznie przez wprawnych legislatorów);
 - o wada: identyfikatory nie odporne na ewentualne zmiany struktury i numeracji jednostek redakcyjnych,
 - o wada: numery jednostek redakcyjnych mogą zawierać formatowanie, które jest znaczące (np. „Art. 17¹” to nie to samo co „Art. 171”), a które może być trudne do zakodowania w treści identyfikatora.

Jeżeli w toku prac legislacyjnych dopuszczalne są zmiany struktury lub numerów jednostek redakcyjnych, wówczas należy zastosować identyfikatory nieznaczące.

3.3 Elementy globalne i lokalne

Elementy, które:

- są wykorzystywane w niezmienionej postaci w kilku miejscach w modelu zawartości (np. `p`),
- mogą być cytowane (np. wszystkie jednostki redakcyjne i systematyzacyjne)

są zadeklarowane jako elementy globalne. Pozostałe elementy są zadeklarowane jako lokalne.

3.4 Typy nazwane

W sytuacji, gdy ten sam model zawartości jest wykorzystywany w elementach o różnych nazwach, jest on zdefiniowany jako typ nazwany (np. `ZawartoscTekstowaTyp`). Deklaracje poszczególnych elementów odwołują się wówczas do tego typu.

Typy nazwane stanowią także bazę do rozszerzeń. Na przykład typ `JednostkaRedakcyjnaTyp` zawiera tylko jeden atrybut `id`. Z tego typu wyprowadzone są typy (nazwane lub anonimowe) wszystkich jednostek redakcyjnych. Dzięki temu rozszerzenie w przyszłości schematu o dodatkowe atrybuty jednostek redakcyjnych będzie wymagało zmiany tylko jednego typu (zob. także informacje o rozbudowie schematów – punkt 6.1).

3.5 Grupy

Nazwane grupy modeli (np. `akapit`) oraz grupy atrybutów (np. `referencja`) są wykorzystywane w sytuacji, gdy pewien fragment modelu zawartości lub grupa atrybutów występuje w modelu wielokrotnie, lecz nie samodzielnie, a jako część składowa bardziej skomplikowanego modelu. Użycie grup pozwala wyeliminować powtarzające się fragmenty deklaracji.

Nazwane grupy modeli oraz grupy atrybutów mogą być także wykorzystane przy rozbudowie schematów (zob. 6.2).

4 Budowa logiczna aktów prawnych

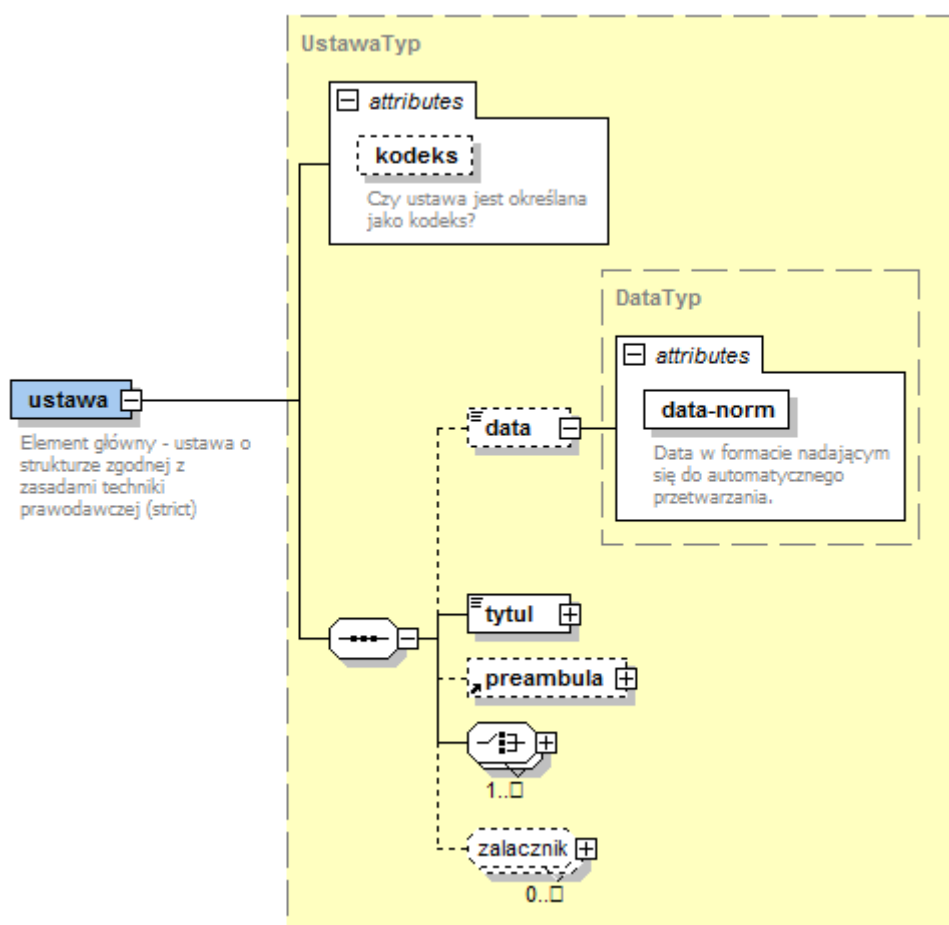
Budowa wszystkich typów dokumentów jest podobna i oparta na tych samych podstawowych konstrukcjach (zawartych w dokumencie schematu `wspolne.xsd`). Różnice pomiędzy wersjami *strict*, *loose* i *base* dotyczą jedynie sposobu zagnieżdżania jednostek redakcyjnych reprezentujących wyliczenia (zob. 4.3.1).

4.1 Nagłówek

4.1.1 Ustawa

Nagłówek ustawy (zarówno w schemacie *strict*, jak i *loose*) składa się z:

- atrybutu `kodeks`, określającego, czy ustawa jest kodeksem; jest to niezbędne, aby poprawnie wizualizować ustępy, które w kodeksach oznacza się paragrafami,
- opcjonalnego elementu `data` (dla projektów aktów prawnych data aktu nie jest określona), zawierającego:
 - o w treści elementu – datę zapisaną słownie (np. „17 września 2012 r.”),
 - o w atrybucie `data-norm` – datę zapisaną jako wartość typu `date` (np. „2012-09-17”);
- tytułu i opcjonalnej preambuły.

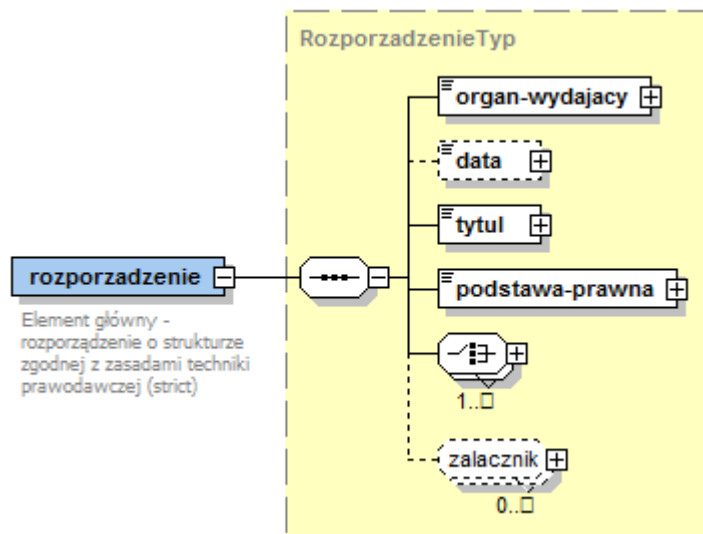


Rysunek 1. Nagłówek ustawy.

4.1.2 Rozporządzenie

Nagłówek rozporządzenia (zarówno w schemacie *strict*, jak i *loose*) składa się z:

- elementu **organ-wydajacy**, zawierającego nazwę organu wydającego,
- opcjonalnego elementu **data** (zob. 4.1.1);
- elementów reprezentujących tytuł rozporządzenia i podstawę prawną.

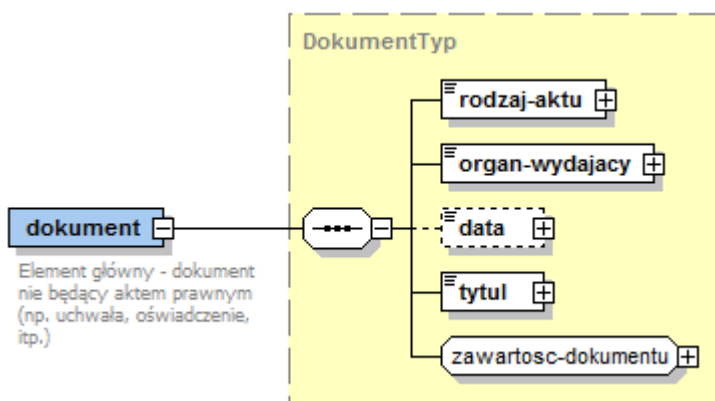


Rysunek 2. Nagłówek rozporządzenia.

4.1.3 Inny dokument

Nagłówek dokumentów zgodnych ze schematem *base* zawiera:

- element **rodzaj-aktu**, zawierający nazwę rodzaju dokumentu (np. „Oświadczenie”),
- element **organ-wydajacy**, zawierający nazwę organu wydającego,
- opcjonalny element **data** (zob. 4.1.1);
- element reprezentujący tytuł dokumentu.



4.2 Jednostki systematyzacyjne

Elementy reprezentujące jednostki systematyzacyjne (część, księga, tytuł, dział, rozdział, oddział) nie tworzą struktury hierarchicznej, lecz zawierają wyłącznie nagłówki jednostek systematyzacyjnych. Treść jednostki systematyzacyjnej (artykuły lub paragrafy) nie jest umieszczana wewnątrz elementu reprezentującego jednostkę systematyzacyjną, lecz jako kolejne elementy na tym samym poziomie.

Przykład:

```

<dzial><nr>I</nr><p>Pory roku</p></dzial>
<rozdzial><nr>1</nr><p>Zima</p></rozdzial>
<artykul><nr>1</nr>
<ustep><nr>1</nr><p>Tworzy się Urząd Kontroli Zimy.</p></ustep>
...
</artykul>

```

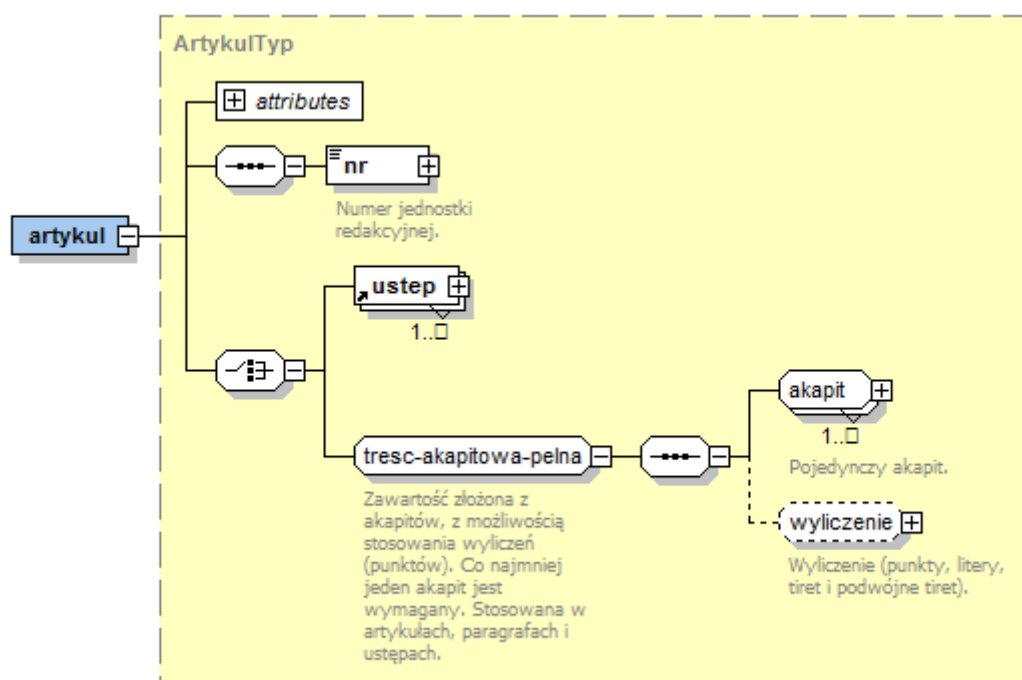
Takie podejście jest uzasadnione tym, że artykuły w ustawie lub paragrafy w rozporządzeniu są numerowane w sposób ciągły. Poszczególne artykuły lub paragrafy można zatem jednoznacznie wskazać wykorzystując ich numery, bez odwoływania się do jednostek systematyzacyjnych. Na przykład ścieżka XPath wskazująca artykuł 37a ustawy ma postać:

```
/ustawa/artykuł[nr="37a"]
```

niezależnie od tego, w której części, księdze, tytule, dziale, rozdziale czy oddziale artykuł ten się znajduje. Gdyby jednostki systematyzacyjne tworzyły strukturę hierarchiczną, przetwarzanie treści aktów byłoby utrudnione.

4.3 Jednostki redakcyjne

Podstawową jednostką redakcyjną ustawy jest artykuł, a rozporządzenia – paragraf. Elementy artykuł i paragraf mają identyczny model zawartości (różny w wersjach *strict*, *loose* i *base*) – zawierają treść akapitową, lub listę ustępów. Ustępy zawierają treść akapitową.



Rysunek 3. Artykuł w schemacie *strict*.

W schemacie *strict* treść akapitowa została zdefiniowana w formie dwóch podobnych grup:

- **tresc-akapitowa** – zawiera dowolnie wiele akapitów (może nie zawierać żadnego) oraz opcjonalne wyliczenie; jest wykorzystywana np. w tabelach, przypisach i innych miejscach, gdzie treść może się zaczynać od razu od wyliczenia i nie musi być poprzedzona akapitem tekstu;
- **tresc-akapitowa-pelna** – zawiera co najmniej jeden akapit oraz opcjonalne wyliczenie; jest wykorzystywana w artykule, paragrafie i ustępie, które nie mogą się zaczynać bezpośrednio od wyliczenia.

Pozostałe schematy (*loose* i *base*) zezwalają, aby artykuł, paragraf czy ustęp zaczynał się od razu od wyliczenia, więc opierają się one na jednej grupie *tresc-akapitowa*.

4.3.1 Wyliczenia

Wyliczenia są reprezentowane przy pomocy elementów *punkt*, *litera*, *tiret* i *podw-tiret*². Różnice pomiędzy schematami *strict*, *loose* i *base* polegają głównie na dopuszczalnym sposobie zagnieżdżania wyliczeń, opisanym poniżej w podpunktach.

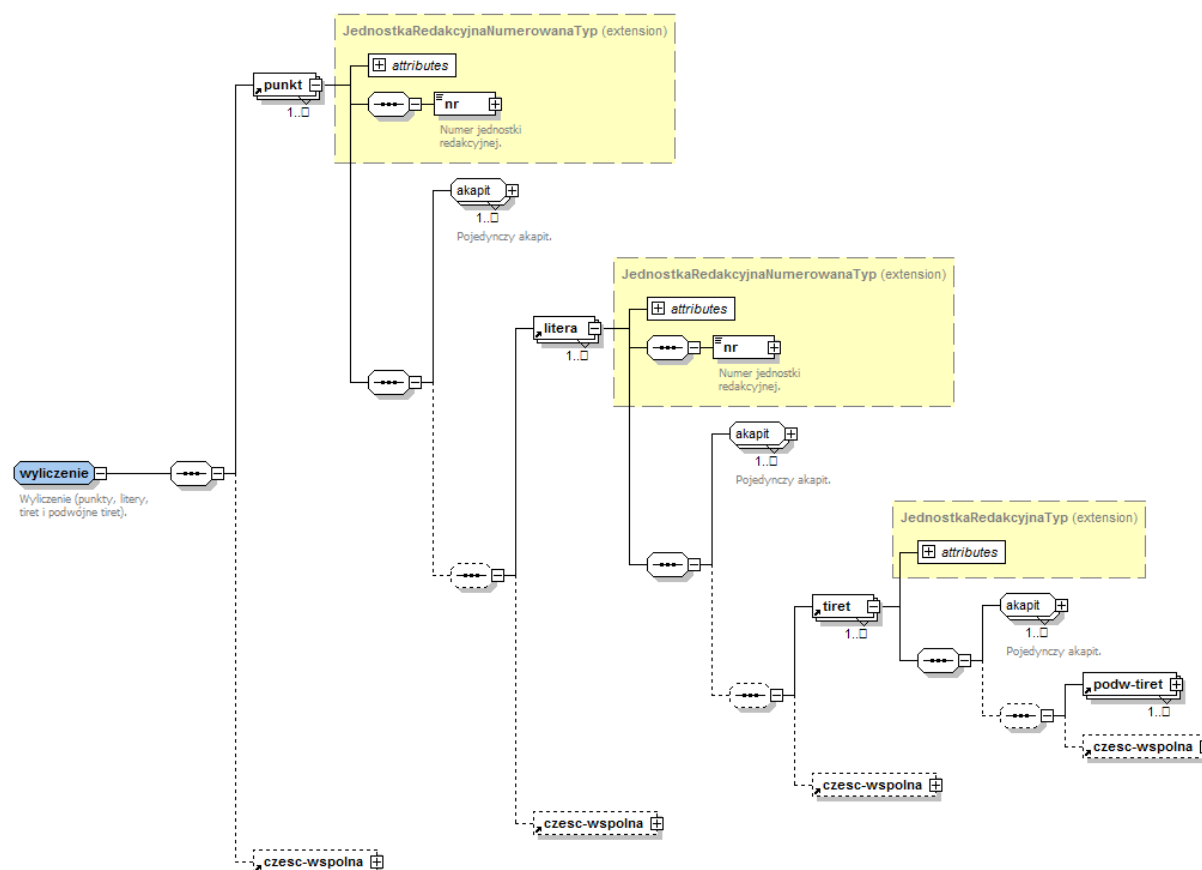
Każdy poziom wyliczenia, we wszystkich schematach, może się kończyć opcjonalną częścią wspólną (element *czesc-wspolna*).

4.3.1.1 Schemat *strict*

Struktura wyliczeń w schemacie *strict* jest ściśle określona:

- punkty zawierają litery,
- litery zawierają tiret,
- tiret zawierają podwójne tiret.

Nie ma możliwości wprowadzenia kolejnych poziomów wyliczenia (głębszych niż podwójne tiret).



Rysunek 4. Wyliczenia w schemacie *strict*.

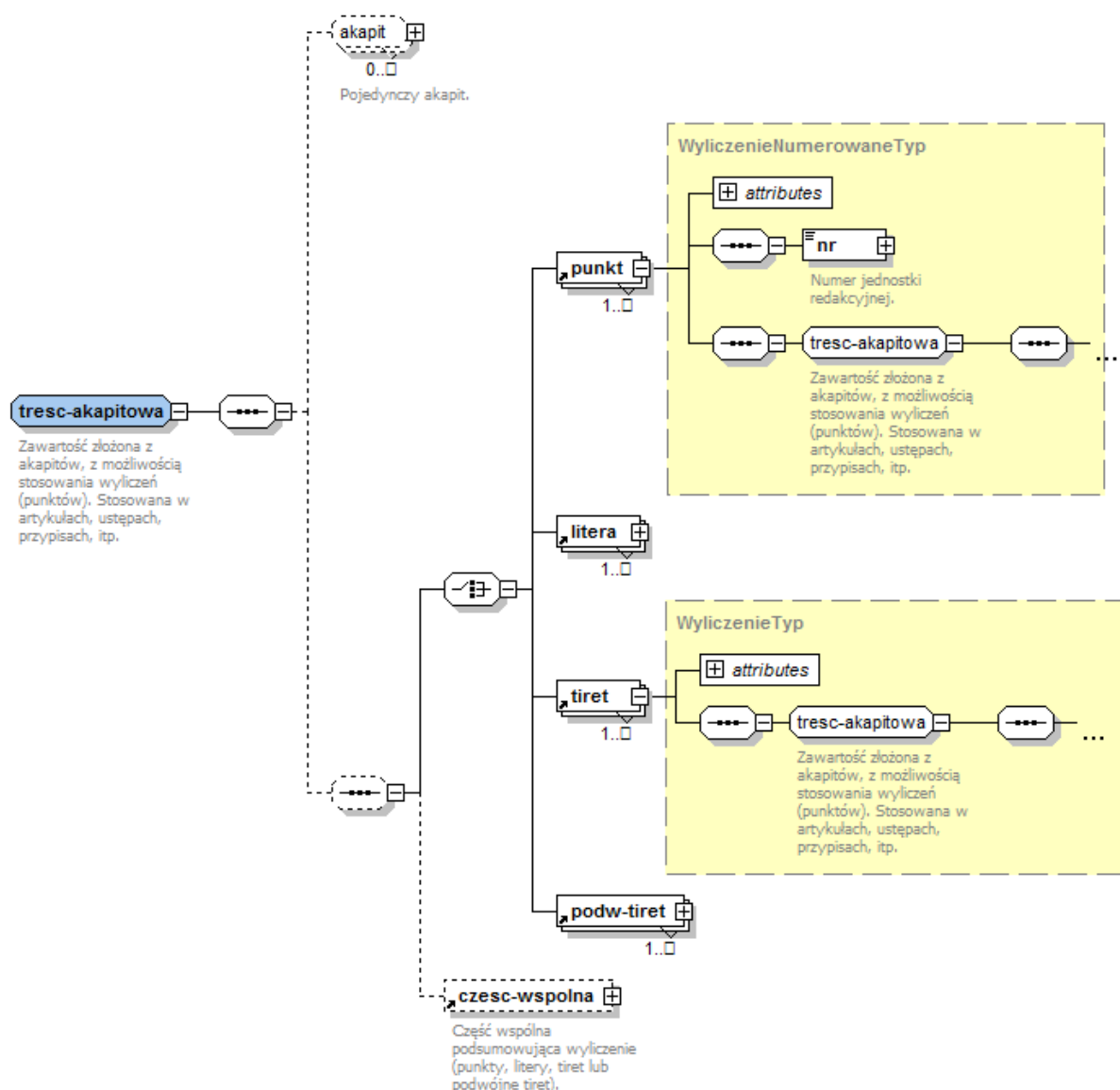
² Podwójne tiret – jednostka redakcyjna nie występująca w zasadach techniki prawodawczej.

4.3.1.2 Schemat *loose*

Schemat *loose* pozwala na umieszczanie elementów wyliczeń (punktów, liter, tiret i podwójnych tiret) na dowolnym poziomie i dowolne ich zagnieżdżanie. W tym celu rekurencyjnie stosowana jest grupa *tresc-akapitowa*.

Elementy *punkt* i *litera* mają ten sam typ *WyliczenieNumerowaneTyp*. Natomiast elementy *tiret* i *podw-tiret* mają typ *WyliczenieTyp*, różniący się od wyliczenia numerowanego tym, że nie ma numeru jednostki redakcyjnej.

W dokumentach zgodnych ze schematem *loose* wyliczenia mogą pojawiać się tylko wewnątrz artykułów (w ustawach) i paragrafów (w rozporządzeniach).

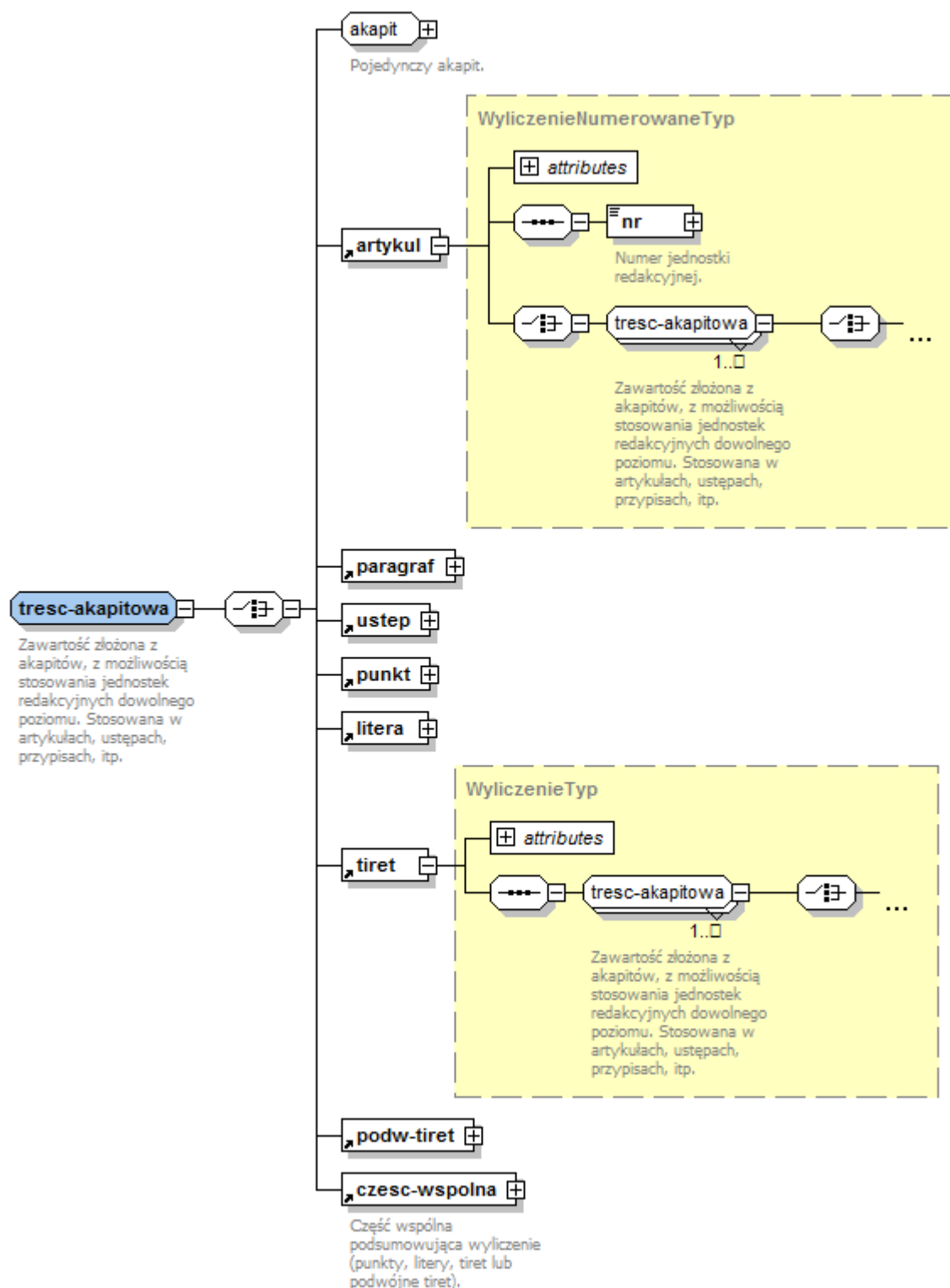


Rysunek 5. Wyliczenie w schemacie *loose*.

4.3.1.3 Schemat *base*

W schemacie *base* elementy reprezentujące wyliczenia (*punkt*, *litera*, *tiret* i *podw-tiret*) są traktowane na równi z artykułami, paragrafami i ustępami – mogą wystąpić bezpośrednio w treści dokumentu i mogą się zagnieżdżać w dowolny sposób. W tym celu – podobnie jak w schemacie *loose* – stosowana jest rekurencyjnie grupa

tresc-akapitowa. Jednak w odróżnieniu od schematu *loose*, grupa ta jest wykorzystywana także w modelu elementów artykuł, paragraf i ustep. Elementy te mają więc – podobnie jak elementy punkt i litera – typ WyliczenieNumerowaneTyp.



Rysunek 6. Wyliczenie w schemacie *base*.

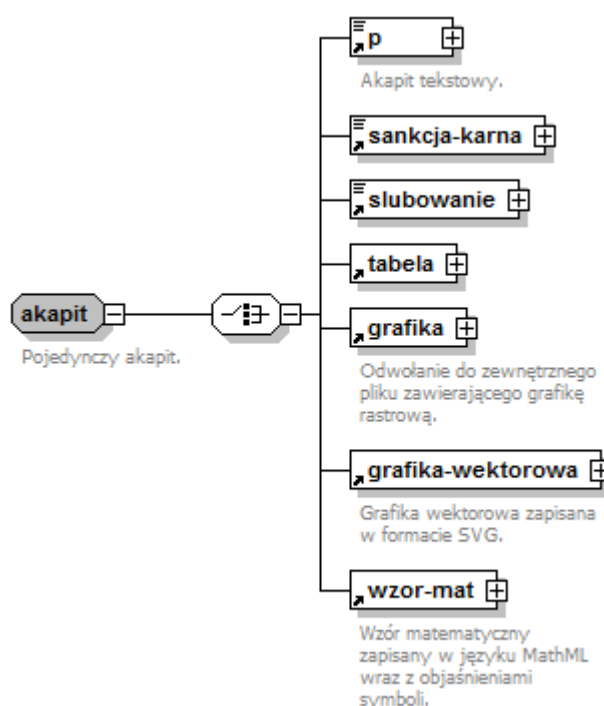
4.3.2 Numery jednostek redakcyjnych

Numery jednostek redakcyjnych (z wyjątkiem tiret i podwójnych tiret, które nie są numerowane) umieszczane są w elemencie `nr`, który jest pierwszym elementem w modelu zawartości jednostki redakcyjnej. Użycie elementów (a nie atrybutów) jest konieczne ponieważ numery jednostek redakcyjnych mogą zawierać podelementy, np.:

- indeks górny (niekiedy występuje numeracja z indeksem górnym, np. 5¹),
- przypisy.

4.4 Akapity

Podstawową jednostką zawartości blokowej, wykorzystywaną we wszystkich jednostkach redakcyjnych, jest akapit. Najczęściej wykorzystywanym rodzajem akapitu jest zwykły akapit tekstowy – element `p`.



Rysunek 7. Akapit.

Szczególnymi rodzajami akapitów są elementy `sankcja-karna` i `slubowanie`.

Wszędzie tam, gdzie można używać akapitów tekstowych, można również umieszczać tabele, grafiki oraz wzory matematyczne.

Wszystkie elementy pełniące rolę akapitów mają atrybut `id` (zob. 3.2), a więc są traktowane jak jednostki redakcyjne. Dzięki temu można się do nich odwoływać np. w aktach nowelizujących.

4.4.1 Tabele

Model tabel używany we wszystkich trzech schematach jest okrojonym i zaadaptowanym modelem tabel z języka HTML. Okrojenie modelu z języka HTML polega na usunięciu atrybutów:

- pozwalających na swobodne formatowanie tabel (w szczególności przy pomocy stylu CSS),
- związanych z obsługą zdarzeń.

Aby umożliwić wygodne sterowanie widocznością ramek tabeli, do modelu dodano następujące atrybuty:

- atrybut `siatka` elementu `tabela`, określający, czy tabela ma widoczną siatkę,
- atrybuty `ramka-lewa`, `ramka-prawa`, `ramka-gorna` i `ramka-dolna`, używane w elementach `col`, `colgroup`, `tr`, `th` i `td`, sterujące widocznością obramowania danej kolumny, grupy kolumn, wiersza lub komórki tabeli.

Odpowiednie ustawienia ramek dla całej tabeli, wiersza, kolumny i pojedynczej komórki są dodawane do siebie. Do wyświetlenia krawędzi pomiędzy dwoma komórkami wystarczy, aby wyświetlanie było efektywnie włączone w choć jednej z tych komórek.

W komórkach tabeli można umieszczać treść akapitową, tj. akapity oraz wyliczenia.

4.4.2 Grafika rastrowa

W trakcie analizy rozpatrywano dwie możliwe techniki umieszczania w aktach prawnych grafiki rastrowej:

- przechowywanie grafiki bezpośrednio w dokumencie XML w formie zakodowanej (Base64 lub szesnastkowo),
- przechowywanie grafiki w osobnym pliku i umieszczanie w dokumencie XML odwołania do tego pliku.

W poniższej tabeli są przedstawione zalety i wady obu rozwiązań:

	Grafika zakodowana w dokumencie XML	Odwołanie do zewnętrznego pliku
Zalety	• wszystko w jednym pliku.	• standardowe rozwiązanie stosowane w technologiach internetowych, • optymalny rozmiar plików, • łatwe przetwarzanie przy pomocy standardowych technologii.
Wady	• drastyczne zwiększenie rozmiaru dokumentu XML, • trudne przetwarzanie – konieczność kodowania i dekodowania, • niemożliwy podgląd przy pomocy standardowych narzędzi (np. przeglądarki internetowej, przekształceń XSLT).	• konieczność posługiwania się wieloma plikami, np. przesyłania dokumentu w formie skompresowanej paczki plików.

Warto zaznaczyć, że przechowywanie grafiki w osobnym pliku nie jest przeszkodą dla podpisu elektronicznego (zob. rozdział 7).

W wyniku powyższej analizy wybrano rozwiązanie, zgodnie z którym grafika rastrowa nie jest przechowywana bezpośrednio w treści dokumentu XML, lecz w osobnym pliku graficznym.

4.4.3 Grafika wektorowa

Grafika wektorowa jest przechowywana bezpośrednio w dokumencie XML w formacie SVG³. Jest to format przetwarzany przez wszystkie wiodące przeglądarki internetowe, więc nie ma potrzeby przechowywania alternatywnej postaci rastrowej takiej grafiki.

Ponieważ schemat XML Schema dla formatu SVG nie jest dostępny, model zawartości elementu `grafika-wektorowa` zawiera symbol wieloznaczny `any`, umożliwiający umieszczenie dowolnego elementu z przestrzeni nazw SVG.

4.4.4 Wzory matematyczne

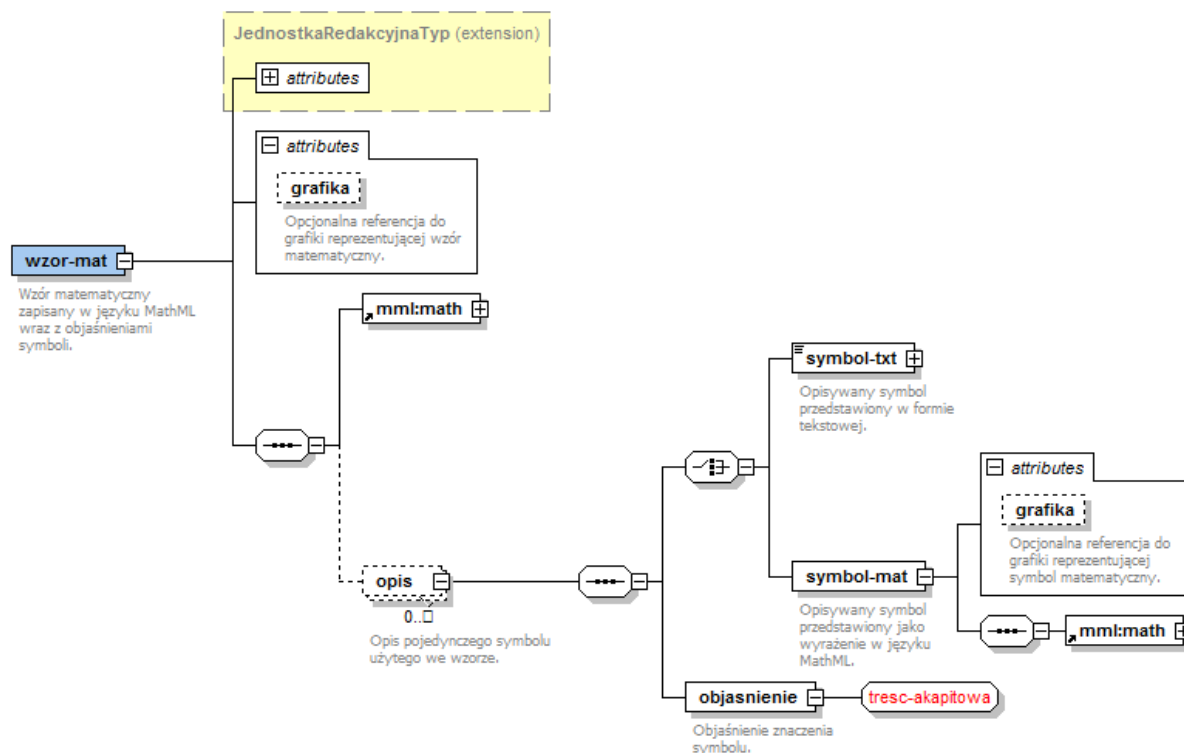
Wzory matematyczne są przechowywane bezpośrednio w dokumencie XML w formacie MathML⁴. Schemat języka MathML 3.0 jest importowany w dokumencie schematu `wspolne.xsd`. Element `wzor-mat` zawiera element `math` należący do przestrzeni nazw języka MathML. Element `math` z kolei zawiera treść wzoru matematycznego.

Język MathML nie jest wspierany przez niektóre przeglądarki internetowe, dlatego alternatywnym sposobem prezentacji wzoru matematycznego jest jego postać graficzna. Element `wzor-mat` posiada opcjonalny atrybut `grafika`, w którym można umieścić odwołanie do zewnętrznego pliku zawierającego graficzną postać wzoru. Wzór zapisany w formacie MathML należy traktować jako źródłowy, natomiast jego postać graficzną – jako dodatkową formę prezentacji.

Pod wzorem matematycznym może być umieszczona opcjonalna legenda – wyjaśnienie użytych symboli. Opisywane symbole mogą być podane w formie tekstowej lub – jeśli są bardziej skomplikowane – w formacie MathML.

³ Zob. <http://www.w3.org/Graphics/SVG/>

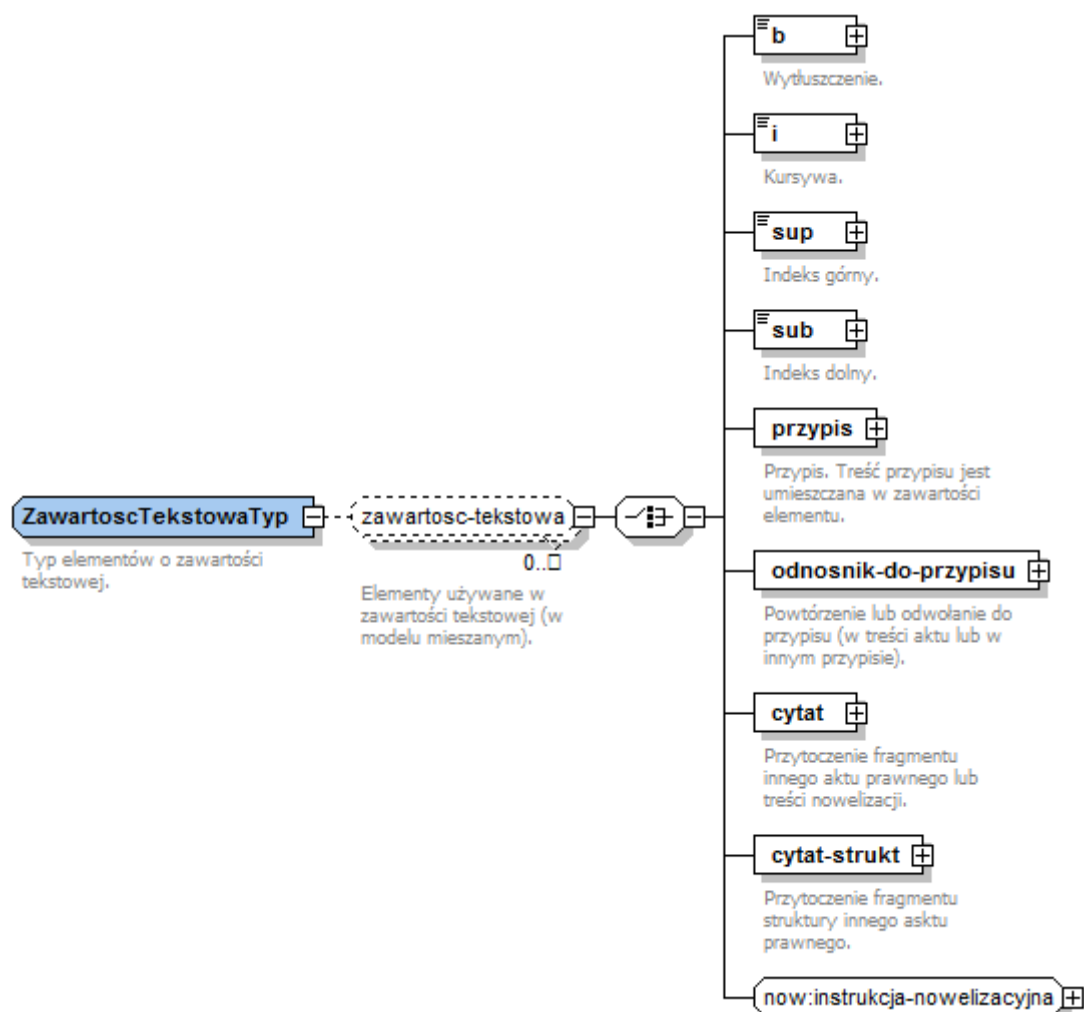
⁴ Zob. <http://www.w3.org/Math/>



Rysunek 8. Wzór matematyczny wraz z objaśnieniami symboli.

4.5 Zawartość tekstowa

Elementy zawierające tekst mają typ `ZawartoscTekstowaTyp`. Jest to model mieszany (ang. *mixed content model*), umożliwiający stosowanie zanurzonych w tekście podelementów, zadeklarowanych w grupie `zawartosc-tekstowa` i opisanych w poniższych podpunktach.



Rysunek 9. Zawartość tekstowa.

4.5.1 Formatowanie

Podstawowe formatowanie można nanosić przy pomocy elementów:

- **b** – wytłuszczenie,
- **i** – kursywa,
- **sup** – indeks górny,
- **sub** – indeks dolny.

4.5.2 Przypisy i odnośniki do przypisów

Element **przypis** umieszczamy w tekście w miejscu, w którym znajduje się odwołanie do przypisu. Treść przypisu umieszczamy wewnątrz elementu **przypis**.

Przypisy są numerowane automatycznie, więc w przypisie nie podajemy jego numeru.

Ze względu na możliwość nowelizowania przypisów, element **przypis** funkcjonuje jako jednostka redakcyjna, a więc posiada atrybut **id** (zob. 3.2).

Element **odnosnik-do-przypisu** należy wykorzystać wówczas, gdy:

- w treści dokumentu znajduje się kilka odwołań do tego samego przypisu (wówczas pierwsze odwołanie należy zapisać jako przypis, a kolejne – jako odnośniki),
- w treści przypisu występuje odwołanie do innego przypisu.

W elemencie `odnosnik-do-przypisu`, w atrybucie `id-przypisu` wskazujemy przypis, do którego się odwołujemy. Każdy przypis, do którego istnieją odnośniki, musi posiadać atrybut `id-przypisu`. Jest to wewnętrzny identyfikator przypisu pozwalający na tworzenie odnośników.

4.5.3 Cytaty

Elementy `cytat` i `cytat-strukt` służą do umieszczania w treści aktu prawnego fragmentów treści innego aktu. Najczęściej elementy te są wykorzystywane (wraz ze znacznikami nowelizacyjnymi – zob. rozdział 5) w aktach nowelizujących do zapisania treści nowelizacji.

W elemencie `cytat` umieszczamy cytat fragmentu przepisu, mieszczący się w ramach otaczającego go akapitu (ang. *inline*) lub cytat zaczynający się od fragmentu tekstowego (*inline*) i zakończony elementami strukturalnymi. W obu przypadkach fragment tekstu cytowany w ramach akapitu umieszczamy w podelemencie `tekst`.

Przykład:

- a) w ust. 5 po wyrazach „ust. 1 pkt 4” dodaje się wyrazy „ , uwzględniając w szczególności:

- 1) założenia ustawy budżetowej,
- 2) przebieg realizacji budżetu państwa.”.

```
<litera><nr>a</nr>
<p>w ust. 5 po wyrazach <cytat><tekst>ust. 1 pkt 4</tekst></cytat>
dodaje się wyrazy <cytat><tekst>, uwzględniając w
szczególności:</tekst>
<punkt><nr>1</nr>założenia ustawy budżetowej,</punkt>
<punkt><nr>2</nr>przebieg realizacji budżetu państwa.</punkt>
</cytat>.</p>
</litera>
```

W elemencie `cytat-strukt` umieszczamy cytat złożony z pełnych jednostek redakcyjnych lub akapitów. Taki cytat jest formatowany w osobnych akapitach. Jednak ze względu na znak przestankowy (przecinek, średnik lub kropkę) występujący bezpośrednio po cytacie i będący logicznie częścią akapitu poprzedzającego cytat, element `cytat-strukt` nie może być traktowany jak akapit, lecz należy do zawartości tekstowej.

Przykład:

- 1) W art. 1 w pkt 3 dodaje się lit. c w brzmieniu:

„c) monitorowania chorób;”;

```
<punkt><nr>1</nr>
<p>W art. 1 w pkt 3 dodaje się lit. c w brzmieniu:
<cytat-strukt>
```

```
<litera><nr>c</nr><p>monitorowania chorób;</p></litera>
</cytat-strukt>;</p>
</punkt>
```

Cytować można dowolne elementy struktury z każdego z trzech schematów. Aby to umożliwić, cytat strukturalny oraz strukturalna część zwykłego cytatu mają model oparty o symbol wieloznaczny any.

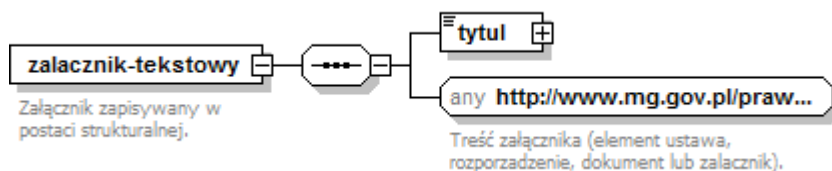
4.5.4 Instrukcje nowelizacyjne

Instrukcje nowelizacyjne są opisane w rozdziale 5.

4.6 Załączniki

Załączniki do aktów prawnych mogą być zapisywane w formie tekstowej lub binarnej. Formę tekstową należy wykorzystywać wówczas, gdy treść załącznika ma postać tekstową, dającą się zapisać zgodnie jednym ze schematów. Formy binarnej należy używać w pozostałych przypadkach (np. formularzy, schematów).

Załączniki tekstowe są w całości umieszczone w treści aktu prawnego. Symbol wieloznaczny any umożliwia wstawienie w elemencie `załącznik-tekstowy` dowolnego elementu z przestrzeni nazw *strict*, *loose* lub *base*. Powinno się za niego podstawić element główny jednego z trzech schematów (tj. *ustawa*, *rozporządzenie* lub *dokument*) lub specjalny element `ba:załącznik`, zadeklarowany w schemacie *base*, umożliwiający umieszczenie jako załącznika dowolnej treści o swobodnej strukturze (zgodnej z modelem *base*, lecz bez nagłówka wymaganego w dokumentach nie będących załącznikami).



Rysunek 10. Załącznik tekstowy.

Przykład:

Jeśli załącznikiem do obwieszczenia o tekście jednolitym jest ustawa nowa (*strict*), wówczas podelementem elementu `załącznik-tekstowy` będzie element `st:ustawa` (gdzie `st` jest prefiksem przestrzeni nazw `http://www.mg.gov.pl/prawo/strict`).

Załączniki binarne są – podobnie jak grafiki rastrowe – przechowywane w zewnętrznych plikach. W akcie prawnym umieszcza się jedynie odsyłacze do plików załączników.

5 Znaczniki nowelizacyjne

Znaczniki nowelizacyjne są instrukcjami umieszczanymi w aktach prawnych zmieniających (nowelizujących) inne akty, umożliwiającymi automatyczne naniesienie zmian w treści aktów zmienianych.

We wszystkich trzech schematach można używać znaczników nowelizacyjnych mających postać atrybutów i elementów należących do odrębnej przestrzeni nazw <http://www.mg.gov.pl/prawo/novelizacje>. Znaczniki nowelizacyjne są bowiem dodatkowymi informacjami i formalnie nie należą do treści aktu prawnego (w szczególności nie powinny być objęte podpisem elektronicznym), a zatem ich semantyczne i składniowe wyróżnienie poprzez odrębną przestrzeń nazw jest uzasadnione. Wykorzystanie odrębnej przestrzeni nazw ułatwia ponadto wyłączenia znaczników nowelizacyjnych spod podpisu elektronicznego (zob. 7.1).

Znaczniki nowelizacyjne nanoszone na fragmenty treści (np. cytaty) zamieszczone w akcie nowelizującym mają postać atrybutów i są zadeklarowane w grupie atrybutów `instr-novelizacyjne`. Dzięki temu znacznik nowelizacyjny nie zaburza struktury aktu prawnego, a jedynie nakłada dodatkową semantykę na istniejące elementy (np. `cytat`, `cytat-strukt`). Takie znaczniki łatwo jest w razie potrzeby usunąć (odfiltrować) czy wyłączyć spod podpisu elektronicznego. Gdyby znacznik nowelizacyjny był elementem zawierającym fragment treści aktu, wówczas jego usunięcie (bez usuwania zawartości) byłoby bardziej skomplikowane.

Najważniejszym atrybutem nowelizacyjnym jest atrybut `funkcja`. Przy jego pomocy określamy, jaką rolę w nowelizacji pełni dany fragment treści. Przy pomocy pozostałych atrybutów określamy precyzyjnie sposób i miejsce naniesienia zmiany. Są one szczegółowo opisane w kolejnych punktach.

Pozostałe znaczniki nowelizacyjne (nie obejmujące fragmentów treści aktu nowelizowanego – np. znaczniki uchylające ustawę lub cały przepis) mają postać elementów o zawartości pustej i są zdefiniowane w grupie `instrukcja-novelizacyjna`. Takie elementy również nie zaburzają struktury aktu prawnego i są łatwe do usunięcia.

W dalszej części tego rozdziału zakładamy, że przestrzeń nazw <http://www.mg.gov.pl/prawo/novelizacje> jest odwzorowana na przedrostek `now`.

5.1 Wskazanie nowelizowanego aktu prawnego

Zasadniczym elementem definiującym nowelizację jest określenie aktu prawnego, który jest nowelizowany. Robimy to umieszczając w elemencie `artykul` lub `paragraf` atrybut `akt`. Wartością tego atrybutu powinien być identyfikator aktu prawnego, który jest nowelizowany w danym artykule lub paragrafie.

Przykład:

Art. 2. W ustawie z dnia 20 lipca 1950 r. o zawodzie felczera (Dz. U. z 2004 r. Nr 53, poz. 531) wprowadza się następujące zmiany:

...

```
<artykul now:akt="DzU/2004/53/531"><nr>2</nr>
<p>W ustawie z dnia 20 lipca 1950 r. o zawodzie felczera (Dz. U. z
2004 r. Nr 53, poz. 531) wprowadza się następujące zmiany:</p>
...
</artykul>
```

Postać identyfikatora aktu prawnego może być dowolna, niekoniecznie zgodna z powyższym przykładem.

Wskazanie nowelizowanego aktu prawnego w artykule lub paragrafie jest konieczne, aby w treści tego artykułu lub paragrafu używać instrukcji nowelizacyjnych opisanych w kolejnych punktach. Jedno takie wskazanie obejmuje wszystkie instrukcje nowelizacyjne umieszczone w treści artykułu lub paragrafu.

5.2 Dodanie jednostek redakcyjnych

Pełne jednostki redakcyjne dodawane w ramach nowelizacji są umieszczane w elemencie `cytat-strukt`. Instrukcja dodania tych jednostek ma postać:

- atrybutu `funkcja` o wartości:
 - o `dodaj-w`, jeśli nowe jednostki dodajemy wewnątrz wskazanej jednostki redakcyjnej, po jej dotychczasowej zawartości,
 - o `dodaj-po`, jeśli nowe jednostki dodajemy po wskazanej jednostce redakcyjnej;
- atrybutu `element`, wskazującego jednostkę redakcyjną nowelizowanego aktu, w której lub po której dodajemy nowe jednostki redakcyjne.

Przykład:

1) W art. 1 w pkt 3 dodaje się lit. c w brzmieniu:

„c) monitorowania chorób;”;

```
<punkt><nr>1</nr>
<p>W art. 1 w pkt 3 dodaje się lit. c w brzmieniu:
<cytat-strukt now:funkcja="dodaj-w" now:element="j331">
<litera><nr>c</nr><p>monitorowania chorób;</p></litera>
</cytat-strukt>;</p>
</punkt>
```

Identyfikator `j331` wskazuje art. 1 pkt 3 nowelizowanego aktu prawnego.

Przykład:

1) W art. 1 w pkt 3 po lit. b dodaje się lit. ba w brzmieniu:

„ba) monitorowania chorób;”;

```
<punkt><nr>1</nr>
<p>W art. 1 w pkt 3 po lit. b dodaje się lit. ba w brzmieniu:
<cytat-strukt now:funkcja="dodaj-po" now:element="j334">
<litera><nr>ba</nr><p>monitorowania chorób;</p></litera>
</cytat-strukt>;</p>
</punkt>
```

Identyfikator `j334` wskazuje art. 1 pkt 3 lit. b nowelizowanego aktu prawnego.

5.3 Dodanie fragmentu przepisu

Dodanie fragmentu przepisu wymaga wskazania:

- fragmentu tekstu, po którym należy dodać nowy fragment,
- dodawanego fragmentu tekstu.

Oba fragmenty umieszczamy w elementach `cytat`, na które nakładamy instrukcję nowelizacyjną:

- w elemencie `cytat`, obejmującym fragment tekstu, po którym należy dodać nowy fragment, umieszczamy:
 - o atrybut `id`, będący identyfikatorem zmiany,
 - o atrybut `funkcja` o wartości `dodaj-po-tekście`,
 - o atrybut `element`, wskazujący jednostkę redakcyjną nowelizowanego aktu, w której ma zostać dodany fragment przepisu,
 - o opcjonalny atrybut `zdanie`, którego wartością jest numer zdania (liczonego pomiędzy kolejnymi znakami kropki), w którym ma zostać dodany fragment przepisu; brak atrybutu oznacza zmianę w treści całej jednostki redakcyjnej,
 - o opcjonalny atrybut `wystapienie`, wskazujący, po którym wystąpieniu fragmentu tekstu należy dodać nowy fragment (dopuszczalne wartości: `pierwsze`, `drugie`, `trzecie`, `ostatnie`, wartość domyślna: `wszystkie`);
- w elemencie `cytat` obejmującym dodawany fragment tekstu umieszczamy:
 - o atrybut `id` o tej samej wartości, co w pierwszym elemencie `cytat`,
 - o atrybut `funkcja` o wartości `nowy-tekst`.

Instrukcja nowelizacyjna jest nakładana na dwa niezależne od siebie elementy `cytat`. Dlatego konieczne jest wskazanie, że te dwa elementy należą do jednej instrukcji nowelizacyjnej. Wskazaniem tym jest atrybut `id` o tej samej wartości w obu elementach `cytat`. Elementy `cytat` opisujące jedną zmianę muszą mieć atrybut `id` o tej samej wartości. Elementy `cytat` opisujące różne zmiany muszą mieć atrybuty `id` o różnych wartościach.

Przykład:

5) w art. 18 w ust. 1 w pierwszym zdaniu po wyrazach „i urzędy dzielnic w gminie Warszawa-Centrum” dodaje się wyrazy „oraz urząd gminy Wesoła”.

```
<punkt><nr>5</nr>
<p>w art. 18 w ust. 1 w pierwszym zdaniu po wyrazach
<cytat now:id="1" now:element="j571" now:funkcja="dodaj-po-tekście"
now:zdanie="1"><tekst>i urzędy dzielnic w gminie Warszawa-
Centrum</tekst></cytat> dodaje się wyrazy <cytat now:id="1"
now:funkcja="nowy-tekst"><tekst>oraz urząd gminy
Wesoła</tekst></cytat>.</p>
</punkt>
```

Identyfikator `j571` wskazuje na art. 18 ust. 1 nowelizowanego aktu prawnego.

5.4 Dodanie fragmentu przepisu i jednostek redakcyjnych

Zmianę polegającą na dodaniu fragmentu tekstu i następujących po nim jednostek redakcyjnych zapisujemy podobnie jak dodanie fragmentu przepisu, przy czym w elemencie `cytat` zawierającym dodawany fragment, po podelemencie `tekst` umieszczamy dodawane jednostki redakcyjne. Atrybuty umieszczane w elementach `cytat` i ich znaczenie są takie same jak w punkcie 5.3. Należy pamiętać, że aby zmiana została poprawnie nałożona na nowelizowany akt, nowy fragment przepisu musi być dodawany na końcu dotychczasowej treści.

Przykład:

a) w ust. 5 po wyrazach „ust. 1 pkt 4” dodaje się wyrazy „ , uwzględniając w szczególności:

- 1) założenia ustawy budżetowej,
- 2) przebieg realizacji budżetu państwa.”.

```
<litera><nr>a</nr>
<p>w ust. 5 po wyrazach <cytat now:id="2" now:element="j442"
now:funkcja="dodaj-po-tekscie"><tekst>ust. 1 pkt 4</tekst></cytat>
dodaje się wyrazy <cytat now:id="2" now:funkcja="nowy-tekst"><tekst>,
uwzględniając w szczególności:</tekst>
<punkt><nr>1</nr>założenia ustawy budżetowej,</punkt>
<punkt><nr>2</nr>przebieg realizacji budżetu państwa.</punkt>
</cytat>.</p>
</litera>
```

Identyfikator j442 wskazuje na ust. 5 nowelizowanego artykułu.

5.5 Uchylenie jednostek redakcyjnych

Uchylenie pełnej jednostki redakcyjnej lub ciągu jednostek zapisuje się przy pomocy elementu `uchyl`, umieszczanego w zawartości tekstowej. Jeżeli uchylamy jedną jednostkę redakcyjną, to jej identyfikator należy podać w atrybucie `element`. Jeżeli uchylamy zakres jednostek redakcyjnych, to identyfikatory pierwszej i ostatniej uchylanej jednostki należy podać w atrybutach `element` i `element-do`.

Przykład:

5) w art. 2 uchyla się pkt 32-34.

```
<punkt><nr>5</nr>
<p><now:uchyl element="j243" element-do="j245"/>w art. 2 uchyla się
pkt 32-34.</p>
</punkt>
```

Identyfikator j243 wskazuje na art. 2 pkt. 32 nowelizowanego aktu prawnego, zaś identyfikator j245 – na art. 2 pkt. 34.

5.6 Uchylenie fragmentu przepisu

Uchylany fragment przepisu jest umieszczany w elemencie `cytat`. Instrukcja uchylenia cytowanego fragmentu ma postać:

- atrybutu `funkcja` o wartości `uchyl-tekst`,
- atrybutu `element`, wskazującego jednostkę redakcyjną nowelizowanego aktu, w której ma zostać uchylony fragment przepisu,
- opcjonalnego atrybutu `zdanie`, którego wartością jest numer zdania (liczonego pomiędzy kolejnymi znakami kropki), w którym ma zostać uchylony fragment przepisu; brak atrybutu oznacza uchylenie w treści całej jednostki redakcyjnej,
- opcjonalny atrybut `wystapienie`, wskazujący, które wystąpienie fragmentu tekstu jest uchylone (dopuszczalne wartości: `pierwsze`, `drugie`, `trzecie`, `ostatnie`, wartość domyślna: `wszystkie`).

Przykład:

3) w art. 49 w ust. 3 skreśla się wyrazy „oraz egzekucją należności pieniężnych”;

```
<punkt nr="3">
<p>w art. 49 w ust. 3 skreśla się wyrazy
<cytat now:element="j142" now:funkcja="uchyl-tekst"><tekst> oraz
egzekucją należności pieniężnych</tekst></cytat>;</p>
</punkt>
```

Identyfikator `j142` wskazuje na art. 49 ust. 3 nowelizowanego aktu prawnego.

5.7 Uchylenie aktu prawnego

Uchylenie całego aktu prawnego zapisuje się przy pomocy elementu `uchyl-akt`, umieszczanego w zawartości tekstowej. W atrybucie `akt` tego elementu należy umieścić identyfikator uchylanego aktu prawnego.

Jeżeli akt prawny jest uchylany z wyjątkiem określonych przepisów, wyjątki te określa się w podelementach `oproc`. Jeżeli wyjątek obejmuje jedną jednostkę redakcyjną, to jej identyfikator należy podać w atrybucie `element`. Jeżeli wyjątek dotyczy zakresu jednostek redakcyjnych, to identyfikatory pierwszej i ostatniej uchylanej jednostki należy podać w atrybutach `element` i `element-do`. Dodatkowo w atrybucie `obowiazuje-do` należy podać datę, do której obowiązuje dany wyjątek.

Przykład:

Art. 10. Uchyla się ustawę z dnia 4 września 1997 r. o funduszach przemysłowych i ich prywatyzacji, z wyjątkiem art. 25-27, które obowiązują do 20 września 2005 r.

```
<artykul><nr>10</nr>
<p><now:uchyl-akt akt="DzU/1997/34/532">
<oproc element="j247" element-do="j265"
obowiazuje-do="2005-09-20"/></now:uchyl-akt>Uchyla się ustawę z dnia 4
września 1997 r. o funduszach przemysłowych i ich prywatyzacji z
```

wyjątkiem art. 25-27, które obowiązują do 20 września 2005 r.</p>
 </artykuł>

Identyfikator j247 wskazuje na art. 25 uchylanej ustawy, zaś identyfikator j265 – na art. 27.

5.8 Zastąpienie jednostek redakcyjnych

Zastępując całe jednostki redakcyjne, podajemy ich nową treść w elemencie `cytat-strukt`. Instrukcja zastąpienia jednostek redakcyjnych ma postać:

- atrybutu `funkcja` o wartości `zastap`,
- atrybutu `element`, wskazującego jednostkę redakcyjną, która będzie zastąpiona,
- opcjonalnego atrybutu `element-do`, wskazującego koniec zakresu jednostek redakcyjnych, które będą zastąpione (atrybut `element` wskazuje wówczas początek zakresu).

Przykład:

2) w art. 2 ust. 1 pkt 1-2 otrzymują brzmienie:

- „1) lekarzy,
- 2) aptekarzy,
- 2a) felczerów,”.

```
<punkt><nr>2</nr>
<p>w art. 2 ust. 1 pkt 1-2 otrzymują brzmienie:
<cytat-strukt now:funkcja="zastap" now:element="j331"
now:element-do="j332">
<punkt><nr>1</nr><p>lekarzy,</p></punkt>
<punkt><nr>2</nr><p>aptekarzy,</p></punkt>
<punkt><nr>2a</nr><p>felczerów,</p></punkt>
</cytat-strukt>.</p>
</punkt>
```

Identyfikator j331 wskazuje na art. 2 ust. 1 pkt 1 nowelizowanego aktu prawnego, zaś identyfikator j332 – na art. 2 ust. 1 pkt 2.

5.9 Zastąpienie fragmentu przepisu

Zastąpienie fragmentu przepisu wymaga wskazania:

- fragmentu tekstu, który ma zostać zastąpiony,
- fragmentu tekstu zastępującego.

Oba fragmenty umieszczamy w elementach `cytat`, na które nakładamy instrukcję nowelizacyjną:

- w elemencie `cytat` obejmującym fragment tekstu, który ma zostać zastąpiony, umieszczamy:
 - o atrybut `id`, będący identyfikatorem zmiany,

- o atrybut `funkcja` o wartości `zastap-tekst`,
- o atrybut `element`, wskazujący jednostkę redakcyjną nowelizowanego aktu, w której ma zostać zastąpiony fragment przepisu,
- o opcjonalny atrybut `zdanie`, którego wartością jest numer zdania (liczonego pomiędzy kolejnymi znakami kropki), w którym ma zostać zastąpiony fragment przepisu; brak atrybutu oznacza zmianę w treści całej jednostki redakcyjnej,
- o opcjonalny atrybut `wystapienie`, wskazujący, które wystąpienie fragmentu tekstu należy zastąpić (dopuszczalne wartości: `pierwsze`, `drugie`, `trzecie`, `ostatnie`, wartość domyślna: `wszystkie`);
- w elemencie `cytat` obejmującym zastępujący fragment tekstu:
 - o atrybut `id` o tej samej wartości, co w pierwszym elemencie `cytat`,
 - o atrybut `funkcja` o wartości `nowy-tekst`.

Instrukcja nowelizacyjna jest nakładana na dwa niezależne od siebie elementy `cytat`. Dlatego konieczne jest wskazanie, że te dwa elementy należą do jednej instrukcji nowelizacyjnej. Wskazaniem tym jest atrybut `id` o tej samej wartości w obu elementach `cytat`. Elementy `cytat` opisujące jedną zmianę muszą mieć atrybut `id` o tej samej wartości. Elementy `cytat` opisujące różne zmiany muszą mieć atrybuty `id` o różnych wartościach.

Przykład:

5) w art. 2 ust. 3 wyraz „armie” zastępuje się wyrazami „siły zbrojne”.

```
<punkt><nr>5</nr>
<p>w art. 2 ust. 3 wyraz <cytat now:id="3" now:element="j571"
now:funkcja="zastap-tekst"><tekst>armie</tekst></cytat> zastępuje się
wyrazami <cytat now:id="3" now:funkcja="nowy-tekst"><tekst>siły
zbrojne</tekst></cytat>.</p>
</punkt>
```

Identyfikator `j571` wskazuje na art. 2 ust. 3 nowelizowanego aktu prawnego.

5.10 Wprowadzenie ustępu

Operacja wprowadzenia ustępu jest w aktach prawnych oznaczana zwrotem typu „w art. X dotychczasową treść oznacza się jako ust. 1 i dodaje się ust. 2”. Operację taką zapisujemy przy pomocy elementu `dodaj-ustep`. W atrybucie `element` tego elementu wskazujemy jednostkę redakcyjną (artykuł w ustawie lub paragraf w rozporządzeniu), której treść zostanie oznaczona jako ustęp.

Po wykonaniu tej operacji należy w normalny sposób dodać kolejne ustępy (zob. 5.2), aby nie pozostawić artykułu lub paragrafu z jednym ustępem.

Przykład:

- 5) w art. 23a dotychczasową treść oznacza się jako ust. 1 i dodaje się ust. 2 w brzmieniu:

„2. Wniesienie odwołania wstrzymuje wykonanie zaleceń pokontrolnych.”;

```
<punkt><nr>5</nr>
<p>w art. 23a dotychczasową treść oznacza się jako ust. 1
<now:dodaj-ustep element="j331"/> i dodaje się ust. 2 w brzmieniu:
<cytat-strukt now:funkcja="dodaj-w" now:element="j331">
<ustep><nr>2</nr><p>Wniesienie odwołania wstrzymuje wykonanie zaleceń
pokontrolnych.</p></ustep>
</cytat-strukt>;</p>
</punkt>
```

Identyfikator j331 wskazuje na art. 23a nowelizowanego aktu prawnego.

5.11 Likwidacja zbędnego ustępu

Operacja likwidacji zbędnego ustępu jest w aktach prawnych oznaczana zwrotem typu „w art. X uchyla się ustęp 1 i oznaczenie ust. 2”. Operację taką zapisujemy przy pomocy elementu `usun-ustep`. W atrybucie `element` tego elementu wskazujemy jednostkę redakcyjną (artykuł w ustawie lub paragraf w rozporządzeniu), zawierający pojedynczy ustęp, którego treść po likwidacji ustępu stanie się treścią artykułu lub paragrafu.

Wykonanie tej operacji należy poprzedzić uchyleniem wszystkich ustępów z wyjątkiem jednego.

Przykład:

- 3) w art. 20 uchyla się ustęp 1 i oznaczenie ust. 2.

```
<punkt><nr>3</nr>
<p><now:uchyl element="j243"/><now:usun-ustep element="j242"/>w art.
20 uchyla się ustęp 1 i oznaczenie ust. 2.</p>
</punkt>
```

5.12 Ręczne zmiany fragmentu przepisu

Niekiedy dodania, uchylenia lub zastąpienia fragmentu przepisu nie da się nanieść na nowelizowany akt automatycznie (dotyczy to np. zastąpienia wyrazów w dowolnym przypadku i liczbie). W takiej sytuacji należy wykorzystać atrybut `funkcja` o wartości `zastap-recznie`, `dodaj-recznie` lub `uchyl-recznie`. Sposób ich wykorzystania jest analogiczny, jak funkcji `zastap-tekst` (zob. 5.9), `dodaj-tekst` (zob. 5.3) i `uchyl-tekst` (zob. 5.6). Użycie wartości `zastap-recznie`, `dodaj-recznie` lub `uchyl-recznie` jest jedynie wskazówką dla mechanizmu nanoszenia nowelizacji, że taką zmianę należy pozostawić do ręcznego wykonania przez użytkownika.

5.13 Zmiany niewidoczne w treści aktu nowelizującego

Jeżeli zmiany opisanej w treści aktu nowelizującego nie da się zapisać przy pomocy konstrukcji opisanych w poprzednich punktach (np. ze względu na jej skomplikowany, niestandardowy opis), wówczas można wykorzystać element `zastap`, przy pomocy którego

opisujemy zastąpienie całej jednostki redakcyjnej przez jej nową postać. Zastąpienie taka nie opiera się na elementach `cytat` ani `cytat-strukt`, i dzięki temu jest niewidoczna w treści aktu nowelizującego.

Jeżeli zastępujemy jedną jednostkę redakcyjną, to jej identyfikator należy podać w atrybucie `element` elementu `zastap`. Jeżeli zastępujemy zakres jednostek redakcyjnych, to identyfikatory pierwszej i ostatniej uchylanej jednostki należy podać w atrybutach `element` i `element-do`. Nową treść wskazanych jednostek redakcyjnych należy umieścić jako zawartość elementu `zastap`.

6 Możliwość rozbudowy schematów

Schematy opisane w niniejszej dokumentacji opisują strukturę projektów aktów prawnych na potrzeby systemu Konsultacje On-line. Mogą więc nie obejmować pewnych struktur danych potrzebnych z punktu widzenia innych instytucji zaangażowanych w proces prawodawczy i zadań przez nie wykonywanych (np. podpisu elektronicznego, historii nowelizacji, itp.) Schematy zostały jednak opracowane w taki sposób, aby możliwa była ich rozbudowa bez konieczności modyfikowania plików źródłowych. Schematy opisywane w niniejszej dokumentacji mogą zatem stanowić bazę rozszerzeń niezbędnych dla innych instytucji.

6.1 Przedefiniowywanie typów

Kluczowe modele zawartości zostały zdefiniowane jako typy nazwane. Dzięki temu możliwe jest ich przedefiniowywanie przy pomocy konstrukcji `redefine`.

Niektóre typy nazwane mają charakter typów bazowych, z których wyprowadzono typy poszczególnych elementów. Należą do nich:

- `JednostkaRedakcyjnaTyp`,
- `JednostkaRedakcyjnaNumerowanaTyp`,
- `ZawartoscTekstowaTyp`.

Przedefiniowanie tych typów spowoduje zmianę w modelu zawartości wszystkich elementów, które mają przypisane te typy oraz typy z nich wyprowadzone.

Przykład:

Przeddefiniowując typ `JednostkaRedakcyjnaTyp`, można do wszystkich elementów traktowanych jako jednostki redakcyjne dodać atrybut oznaczający status jednostki redakcyjnej:

```
<xs:redefine schemaLocation="strict.xsd">
  <xs:complexType name="JednostkaRedakcyjnaTyp">
    <xs:complexContent>
      <xs:extension base="JednostkaRedakcyjnaTyp">
        <xs:attribute name="status" type="StatusTyp"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:redefine>
```

Analogicznie można przeddefiniowywać pozostałe typy nazwane. Na przykład rozszerzając typy `UstawaTyp`, `RozporządzenieTyp` i `DokumentTyp`, można dodać do poszczególnych typów dokumentów elementy przechowujące podpisy elektroniczne.

6.2 Przeddefiniowywanie grup

Wielokrotnie wykorzystywane fragmenty modeli zawartości są zadeklarowane jako nazwane grupy modeli (ang. *named model groups*), zaś wielokrotnie wykorzystywane zestawy atrybutów – jako grupy atrybutów (ang. *attribute groups*). Grupy te można przeddefiniowywać, zmieniając w ten sposób model zawartości we wszystkich odwołaniach do grup.

Przykład:

Przeddefiniowując grupę `akapit`, można do niej dodać kolejne rodzaje akapitów:

```
<xs:redefine schemaLocation="strict.xsd">
  <xs:group name="akapit">
    <xs:choice>
      <xs:group ref="akapit"/>
      <xs:element name="film" type="MultimediaTyp"/>
      <xs:element name="utwor-muz" type="MultimediaTyp"/>
    </xs:choice>
  </xs:group>
</xs:redefine>
```

7 Wykorzystanie podpisu elektronicznego

Akty prawne są przed publikacją podpisywane podpisem elektronicznym. Istotnym wymaganiem stawianym projektowanym schematom jest możliwość wyłączenia spod podpisu elektronicznego znaczników nowelizacyjnych, nie stanowiących normatywnej treści aktu. Ponadto należy zapewnić możliwość podpisywania wraz z aktem prawnym grafik rastrowych i załączników binarnych, umieszczanych w odrębnych plikach. Zagadnienia te są omówione w poniższych punktach.

7.1 Wyłączanie znaczników nowelizacyjnych spod podpisu elektronicznego

Znaczniki nowelizacyjne mają postać elementów o zawartości pustej oraz atrybutów należących do odrębnej przestrzeni nazw. Dzięki temu możliwe jest ich łatwe wyłączenie spod podpisu elektronicznego.

Wykorzystując mechanizm XML Signature Syntax and Processing⁵, znaczniki nowelizacyjne można wyłączyć spod podpisu elektronicznego, stosując transformację XPath filtering, której parametr XPath jest określony następująco:

```
not(@now:*|ancestor-or-self::now:*)
```

gdzie `now` jest prefiksem przestrzeni nazw

<http://www.mg.gov.pl/prawo/nowelizacje>.

⁵ XML Signature Syntax and Processing (Second Edition), W3C Recommendation 10 June 2008, <http://www.w3.org/TR/xmlsig-core/>

7.2 Podpisywanie zewnętrznych plików binarnych

Zewnętrzne pliki binarne (grafiki rastrowe oraz załączniki binarne) można podpisywać wraz z głównym aktem prawnym w następujący sposób:

- obliczamy skrót każdego z plików binarnych (przy pomocy funkcji skrótu stosowanej do podpisu elektronicznego),
- skrót każdego pliku binarnego umieszczamy w akcie prawnym np. jako dodatkowy atrybut w elemencie zawierającym odwołanie do pliku binarnego,
- podpisujemy akt prawny wraz ze skrótami plików binarnych.

Weryfikacja podpisu w odniesieniu do plików binarnych polega na porównaniu skrótu zapisanego w podpisanym akcie ze skrótem obliczonym ponownie dla pliku binarnego.

Schematy opisane w niniejszej dokumentacji nie zawierają atrybutów pozwalających na umieszczanie skrótów plików binarnych. Możliwe jest jednak rozszerzenie tych schematów poprzez zdefiniowanie grupy atrybutów `referencja`, która jest wykorzystywana we wszystkich odwołaniach do plików binarnych. Grupa ta zawiera jedynie atrybut `href`. Zdefiniowanie powinno polegać na dodaniu do grupy atrybutu, w którym będzie przechowywany skrót, np.:

```
<xs:redefine schemaLocation="strict.xsd">
  <xs:attributeGroup name="referencja">
    <xs:attributeGroup ref="referencja"/>
    <xs:attribute name="skrot" type="xs:string"/>
  </xs:attributeGroup>
</xs:redefine>
```

8 Porównanie z innymi rozwiązaniami

8.1 EDAP

Schemat wykorzystywany w Edytorze Aktów Prawnych EDAP Ministerstwa Spraw Wewnętrznych⁶ był przygotowany na potrzeby polskiego procesu legislacyjnego i w związku z tym najbardziej odpowiada w zakresie nazewnictwa elementów polskim potrzebom. Jednocześnie nie opiera się on na powszechnie stosowanym w schematach aktów prawnych podziale na elementy budujące strukturę hierarchiczną (jednostki systematyzacyjne i redakcyjne), akapity oraz zawartość tekstową. Brak takiego podziału, przejawiający się np. użyciem modelu mieszanego w elementach takich jak artykuł, ustęp, punkt, należy uznać za błąd w sztuce.

W poniższej tabeli jest przedstawione szczegółowe porównanie schematów opisanych w niniejszej dokumentacji ze schematem EDAP.

EDAP	Schematy opisane w niniejszej dokumentacji
Jeden dokument schematu.	Rozbicie na kilka dokumentów schematów

⁶ Plik `schemat.xsd`, będący załącznikiem nr 1 do załącznika nr 3 do umowy na wykonanie ujednoliconego standardu przechowywania i wymiany plików w procesie legislacyjnym.

	(zob. 2.3).
Jedna przestrzeń nazw.	Osobne przestrzenie nazw <i>strict</i> , <i>loose</i> i <i>base</i> . Odrębna przestrzeń nazw dla znaczników nowelizacyjnych (zob. 2.2).
14 typów dokumentów (ustawa, rozporządzenie, uchwała, zarządzenie, informacja, komunikat, decyzja, oświadczenie, wyrok TK, postanowienie, wyrok, porozumienie, ogłoszenie, obwieszczenie).	3 typy dokumentów: ustawa, rozporządzenie i dokument (dla wszystkich pozostałych rodzajów dokumentów) (zob. 2.1).
Brak rozróżnienia na akty nowe i archiwalne oraz bardziej precyzyjnej kontroli struktury aktów nowych.	Rozróżnienie na akty nowe i archiwalne. Precyzyjna kontrola struktury aktów nowych.
Zawiera atrybuty określające status aktu prawnego i poszczególnych jednostek redakcyjnych (uchylony, status, komentarz, widoczny, obowiązuje - od, obowiązuje - do) oraz elementy związane z obowiązywaniem aktu prawnego (trybunał, przepisy - wykonawcze, wejście - w życie, stosowanie).	Schematy projektów aktów prawnych – nie zawierają atrybutów ani elementów określających status aktu prawnego i związanych z jego obowiązywaniem. Niektóre informacje przechowywane w dokumencie EDAP (np. powiązania z aktami wykonawczymi, orzeczeniami sądów) będą dostępne na poziomie systemu Konsultacje On-line. Wymienione atrybuty i elementy mogą być dodane do schematów w ramach rozszerzeń (zob. rozdział 6).
Zawiera uwagi i komentarze.	Nie zawiera uwag ani komentarzy. Uwagi i komentarze będą wprowadzane i zarządzane na poziomie systemu Konsultacje On-line.
Zawiera wyspy danych, umożliwiające zapis dowolnego fragmentu tekstu wykraczającego strukturą bądź formatowaniem poza schemat. Nie zawiera odrębnych struktur dla grafik ani wzorów matematycznych.	Nie zawiera wysp danych. Zawiera mechanizm pozwalający na umieszczanie w treści aktu grafik rastrowych i wektorowych oraz wzorów matematycznych (zob. 4.4).
Zawiera podpisy i informacje o zatwierdzeniu aktu.	Nie zawiera podpisów ani informacji o zatwierdzeniu aktu.
Hierarchiczna struktura jednostek systematyzacyjnych (jednostka systematyzacyjna zawiera jednostki podrzędne oraz artykuły/paragrafy).	Brak hierarchicznej struktury jednostek systematyzacyjnych, płaska kolekcja artykułów/paragrafów i nagłówków jednostek systematyzacyjnych (zob. 4.2).
Nadużywanie modelu mieszanego w jednostkach redakcyjnych (artykułach, ustępach, punktach, itp.) – schemat dopuszcza np. umieszczanie swobodnego	Model mieszany stosowany tylko tam, gdzie mamy do czynienia z zawartością tekstową, i tylko w odniesieniu do elementów formatowanych w ramach otaczającego je

tekstu pomiędzy kolejnymi ustępami w artykule.	akapitu (ang. <i>inline</i>) (zob. 4.5). Jednostki redakcyjne mają model elementowy (zob. 4.3).
Elementy używane wewnątrz akapitów (ang. <i>inline</i>) wymieszane z elementami blokowymi (akapitami) i jednostkami redakcyjnymi niższego poziomu (punktami, literami, tiret).	Model zawartości tekstowej (typ <code>ZawartoscTekstowaTyp</code>) jasno i czytelnie oddzielony od modelu blokowego (grupa <code>akapit</code>) oraz od hierarchii jednostek redakcyjnych.
Dopuszczalne pomieszanie poziomów jednostek redakcyjnych, np. umieszczenie w artykule kilku ustępów, po nich punktów, podwyliczenia (części wspólnej), a następnie liter i tiret (wymienione elementy występują w sekwencji).	W schematach <i>strict</i> i <i>loose</i> brak możliwości pomieszania poziomów jednostek redakcyjnych – na danym poziomie można umieścić jednostki redakcyjne jednego rodzaju.
Treść przypisów zgrupowana w końcowej części dokumentu; w treści dokumentu jedynie odwołania do przypisów. Numeracja przypisów w treści dokumentu.	Treść przypisów w miejscu wystąpienia odwołania. Automatyczna numeracja przypisów w ramach wizualizacji (tj. brak numeru przypisu w dokumencie) (zob. 4.5.2).
Znaczniki nowelizacyjne w formie elementów „wbudowanych” w strukturę aktu, otaczających tekst nowelizacji.	Znaczniki nowelizacyjne wyodrębnione semantycznie i składniowo w odrębnej przestrzeni nazw, nakładane w sposób nie zaburzający struktury aktu, łatwe do wykluczenia spod podpisu elektronicznego lub usunięcia (zob. rozdział 5).
Nieokreślone typy danych w deklaracjach wielu elementów i atrybutów.	Precyzyjnie określone typy elementów i atrybutów.
Własne elementy formatujące (wytłuszczenie, kursywa) i model tabel.	Standardowe elementy formatujące i model tabel bazują na języku HTML.

8.2 CEN MetaLex

CEN MetaLex⁷ jest formatem wymiany dokumentów legislacyjnych, „najmniejszym wspólnym mianownikiem” różnych standardów zapisu aktów prawnych, który nie ma na celu zastąpić lokalnych standardów. Jest więc siłą rzeczy standardem bardzo uniwersalnym. Jednocześnie umożliwia (a wręcz zaleca) tworzenie własnych schematów, dostosowanych do lokalnej specyfiki, zgodnych ze standardem CEN MetaLex.

W poniższej tabeli jest przedstawione porównanie schematów opisanych w niniejszej dokumentacji ze standardem CEN MetaLex.

⁷ Zob. <http://metalex.eu/>; do porównania użyto schematu MetaLex CEN main schema Release 3/05/2008 oraz dokumentacji „CEN Workshop Agreement CWA 15710”, February 2010.

CEN MetaLex	Schematy opisane w niniejszej dokumentacji
Jeden typ dokumentu do ogólnego zastosowania (root)	Pięć typów dokumentów: <i>ustawa</i> (odmiana <i>strict</i> i <i>loose</i>), <i>rozporządzenie</i> (odmiana <i>strict</i> i <i>loose</i>) i <i>dokument</i> .
Jedynie podstawowe, ogólne elementy (container , hcontainer , block , inline).	Konkretne elementy odpowiadające strukturom stosowanym w polskich aktach prawnych.
Bardzo duże możliwości rozszerzeń: cały model oparty na typach bazowych, które można przeddefiniowywać i/lub wyprowadzać z nich własne typy, oraz na grupach zamiennych.	Ograniczone możliwości rozszerzeń (zob. rozdział 6).
Bogaty zbiór metadanych, zdefiniowany w formie ontologii w języku OWL (dotyczących m.in. obiektów bibliograficznych, źródeł prawa, zdarzeń legislacyjnych i ich uczestników, wersji aktów prawnych, itp.)	Brak rozbudowanych metadanych. Niektóre metadane i powiązania będą dostępne na poziomie systemu Konsultacje On-line.
Brak bezpośredniego wsparcia dla grafiki, wzorów matematycznych, tabel.	Wsparcie dla grafiki, wzorów matematycznych, tabel.

8.2.1 Możliwość konwersji do formatu CEN MetaLex

Dzięki jasnemu i czytelnemu podziałowi na strukturę hierarchiczną, akapity i zawartość tekstową, jaki jest zastosowany w schematach opisanych w niniejszej dokumentacji, możliwa jest ich łatwa transformacja do formatu CEN MetaLex, przynajmniej w podstawowym zakresie tekstowym. Transformacja taka powinna polegać na przekształceniu elementów reprezentujących strukturę hierarchiczną, akapity i zawartość tekstową na odpowiednie uniwersalne elementy CEN MetaLex (**container**, **hcontainer**, **block**, **inline**, itd.)

Następujące konstrukcje mogą sprawiać problemy przy transformacji:

- grafiki wektorowe w formacie SVG i wzory matematyczne w formacie MathML – standard CEN MetaLex nie zawiera wsparcia dla grafik wektorowych i wzorów matematycznych; wydaje się, że należy je przekształcać na element typu `absQuotationType`, zezwalający na dowolną zawartość należącą do dowolnej przestrzeni nazw,
- grafiki rastrowe i załączniki binarne, mające postać referencji do zewnętrznych plików – standard CEN MetaLex nie zawiera wsparcia dla grafik i załączników binarnych; wydaje się, że do wskazania zewnętrznego pliku należy stosować standardowy atrybut `metalex:src`,

- tabele – standard CEN MetaLex nie zawiera wsparcia dla tabel, zaś jego dokumentacja nie wskazuje, jak należy sobie radzić z tabelami; rozsądnym rozwiązaniem wydaje się potraktowanie wierszy tabeli jako akapitów (element `block` lub pochodne), zaś komórek – jako zawartości tekstowej (element `inline` lub pochodne),
- cytaty należące do zawartości tekstowej, lecz wprowadzające wewnątrz cytatu strukturę hierarchiczną i akapity – trudno jest wskazać właściwe rozwiązanie, ponieważ schemat CEN MetaLex jest w tym zakresie niekompletny i niezgodny z dokumentacją⁸.

8.3 Akoma Ntoso

Akoma Ntoso⁹ jest standardem bardzo rozbudowanym, obejmującym wiele typów dokumentów oraz danych nie występujących w schematach opisanych w niniejszej dokumentacji. Jest też z definicji rozwiązaniem bardzo ogólnym, dopuszczającym dowolne zagnieżdżenia elementów hierarchii i nie ma w nim ścisłej kontroli struktury, więc byłoby go trudno wykorzystać bez adaptacji (ograniczenia nadmiernej ogólności) w polskich warunkach. Struktura aktu prawnego i rozwiązania zastosowane w schemacie Akoma Ntoso (w szczególności czytelny podział na strukturę hierarchiczną, akapity i zawartość tekstową) są jednak bardzo podobne do stosowanych w schematach opisanych w niniejszej dokumentacji. Dlatego ewentualna konwersja dokumentów zgodnych ze schematami opisanymi w niniejszej dokumentacji do formatu Akoma Ntoso nie powinna sprawić problemów.

W poniższej tabeli jest przedstawione porównanie schematów opisanych w niniejszej dokumentacji ze schematem Akoma Ntoso.

Akoma Ntoso	Schematy opisane w niniejszej dokumentacji
Wiele typów dokumentów dla różnych zastosowań (np. <code>act</code> , <code>officialGazette</code> , <code>debate</code> , <code>debateReport</code> , <code>amendment</code> , <code>judgement</code>)	Podstawowe typy aktów prawnych (<code>ustawa</code> i <code>rozporządzenie</code>) oraz uniwersalny schemat <code>base</code> dla dokumentów innego rodzaju.
Rozbudowane metadane (m. in. zdarzenia, workflow, słowa kluczowe, powiązania z innymi aktami prawnymi, powiązania z elementami ontologii, np. osoba, organizacja, pojęcie, zdarzenie, lokalizacja, itp.)	Brak rozbudowanych metadanych. Niektóre metadane i powiązania będą dostępne na poziomie systemu Konsultacje On-line.
Model zawartości tekstowej (ang. <i>inline</i>) jasno i czytelnie oddzielony od modelu blokowego	

⁸ Dokumentacja (punkt 3.1.1) wskazuje, że typ `absQuotationType` jest typem stosowanym w zawartości tekstowej (*inline*), zawierającym fragment innego dokumentu. Jednak ten typ nie jest nigdzie w schemacie CEN MetaLex wykorzystywany. Element `quotation`, wspomniany w dokumentacji, nie występuje w schemacie. Z typu `absQuotationType` są wyprowadzone jego dwa nieudokumentowane ograniczenia: `quoteStructureType` i `quoteTextType`. Nazwa pierwszego z nich każe się domyślać, że służy on do cytowania struktury, czemu jednak przeczy jego pusta zawartość, nie dopuszczająca ani zawartości tekstowej, ani podelementów, a jedynie atrybuty.

⁹ Zob. <http://www.akomantoso.org/>; do porównania użyto schematu Akoma Ntoso 2.0 Release 15/07/2011.

(akapitów) oraz od hierarchii jednostek redakcyjnych.	
Struktura hierarchiczna aktu oparta na kilkunastu dowolnie zagnieżdżanych elementach (<i>clause</i> , <i>section</i> , <i>part</i> , <i>paragraph</i> , itd.).	Swobodne zagnieżdżanie na podobnej zasadzie jest wykorzystane w schemacie <i>base</i> . Schematy <i>strict</i> i <i>loose</i> nakładają ograniczenia na dopuszczalne sposoby zagnieżdżania.
Element hierarchii zawiera bloki tekstu (element <i>content</i>) lub zagnieżdżone elementy hierarchii, poprzedzone wstępem i zakończone podsumowaniem (częścią wspólną).	Podobna budowa elementów hierarchii (uwzględniająca dodatkowe ograniczenia wynikające z polskiej specyfiki).
Wykorzystanie standardowych elementów języka HTML.	
Bardzo rozbudowany model zawartości tekstowej (ang. <i>inline</i>), zawierający m. in. uwagi odnośnie przebiegu debaty (np. aplauz, śmiechy), wskazania występujących osób, głosowań, oznaczenia elementów ontologii (osób, organizacji, zdarzeń, pojęć, lokalizacji), wskazania stron rozprawy sądowej, uczestniczących w niej prawników, wygłaszanych opinii, itp.	Ograniczony model zawartości tekstowej (zob. 4.5).
Uniwersalny element <i>foreign</i> , umożliwiający umieszczanie w dokumencie struktur spoza schematu AkomaNtoso, np. wzorów matematycznych, grafik wektorowych w formacie SVG.	Specjalne elementy przeznaczone do umieszczania wzorów matematycznych i grafik wektorowych w formacie SVG (zob. 4.4.3, 4.4.4.)
Grafiki rastrowe jako odrębne pliki; w treści aktu odwołanie do grafiki.	
Możliwości dostosowywania (ograniczania lub rozbudowywania) schematu.	
Uniwersalne elementy na poziomie struktury hierarchicznej, akapitów oraz zawartości tekstowej, pozwalające na łatwe rozszerzanie standardu bez rozbudowywania schematu, np.: <pre><inline name="firstName">Jan</inline></pre>	Brak elementów uniwersalnych. Możliwość dodawania specyficznych elementów w ramach rozbudowy schematów (zob. rozdział 6).

8.4 Norme In Rete

Nie można dokonać porównania schematów opisanych w niniejszej dokumentacji z projektem Norme In Rete, ponieważ w Internecie nie jest dostępny schemat stosowany w ramach tego projektu ani jakakolwiek dokumentacja w języku angielskim.