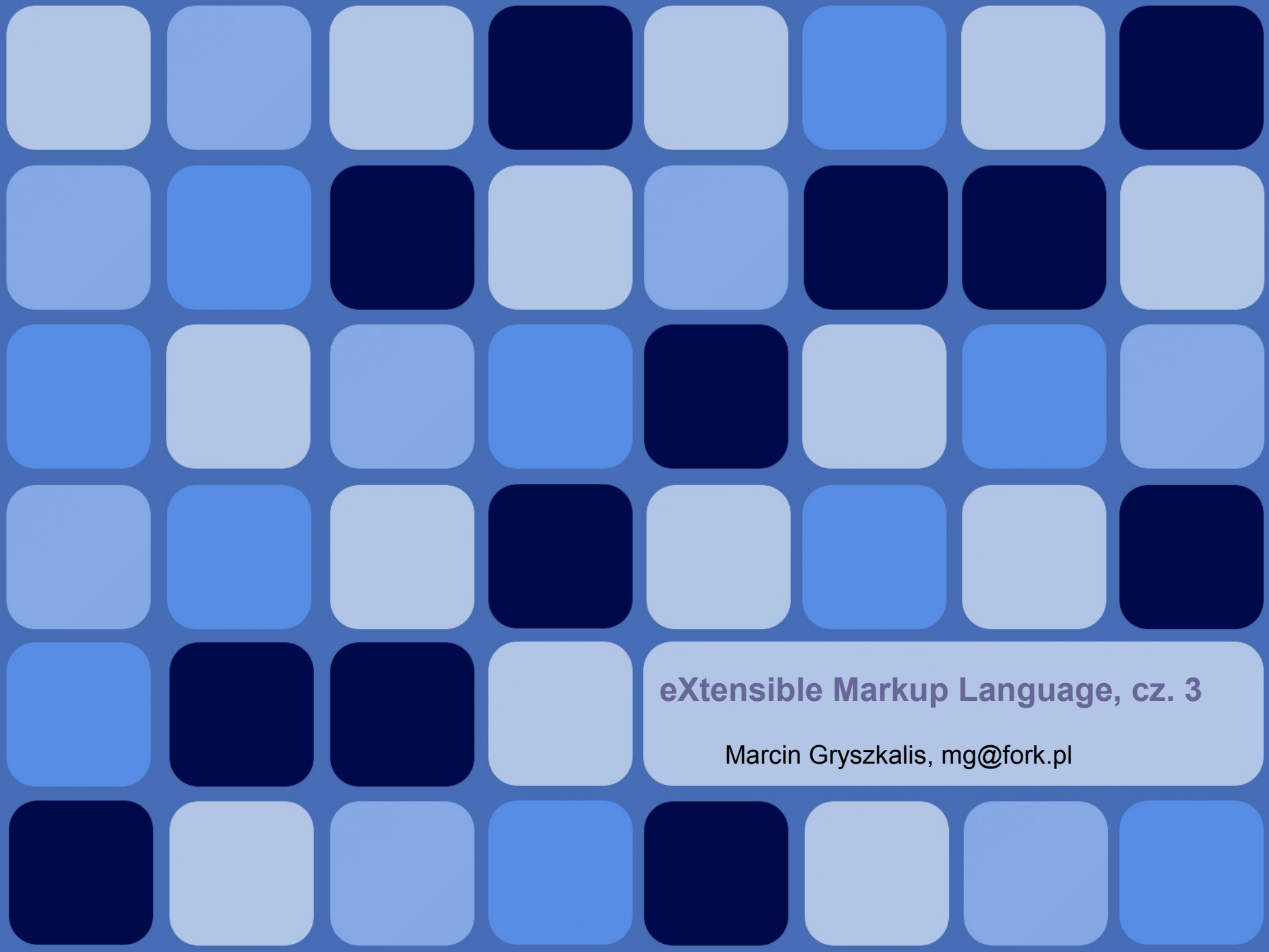

eXtensible Markup Language, cz. 3

Marcin Gryszkalis, mg@fork.pl

XML 3 - Poprawność strukturalna

- Dokument XML musi być poprawny pod względem strukturalnym o ile struktura została zdefiniowana
- Istnieje wiele metod definiowania struktury
  - DTD, Document Type Definition
  - XML Schema (standard W3C, następca DTD)
  - RELAX NG i ISO DSDL (standard ISO, OASIS)
- W specyfikacji oryginalnej: *valid*
- Poprawność strukturalna NIE jest warunkiem koniecznym dla dopuszczenia dalszego przetwarzania dokumentu XML

XML 3 – motywacja

- Opis struktury dokumentu (np. DTD) określa jakie elementy mogą występować w dokumencie oraz jakie atrybuty mogą być przypisane elementom
- Dzięki takiemu opisowi możliwe są:
  - skuteczniejsza kontrola błędów przy tworzeniu
  - implementacja podpowiedzi w edytorach
  - uzgodnienie formatów dla elektronicznej wymiany dokumentów
- Definiowanie języka przez przykład jest niejednoznaczne

XML 3 – DTD – cechy

- Dokument XML nie musi zawierać odwołania do DTD
- Założenie: wszystko co nie jest dozwolone jest zakazane
- Deklaracje DTD mogą być zawarte w pliku XML lub dołączane z zewnętrznego pliku za pomocą odpowiedniego wyrażenia
- DTD nie ma struktury XML-a (pozostałość po SGML)
- Mozna używać komentarzy takich jak w XML-u

XML 3 – definicja DTD

- definicja wewnętrzna – pojedynczy blok DOCTYPE wewnątrz dokumentu XMLowego:

```
<!DOCTYPE nazwa[
        <!ELEMENT nazwa_el_glownego (#PCDATA) >
]>
```

- definicja zewnętrzna:

```
<!DOCTYPE nazwa SYSTEM “innyplik.dtd” >
<!DOCTYPE nazwa SYSTEM “http://a.b.c/plik.dtd” >
```

- dla jednocześnie użytej definicji wewnętrznej i zewnętrznej wyższy priorytet mają deklaracje wewnętrzne (przysłaniają te z definicji zewnętrznej)

XML 3 – definicja DTD cd

- definicja publiczna – po słowie kluczowym PUBLIC występuje dodatkowo tzw. FPI (*formal public identifier*), np.:

  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">

  <!DOCTYPE rss PUBLIC
  “-//Netscape Communications//DTD RSS 0.91//EN”
  “http://my.netscape.com/publish/formats/rss0.91. dtd”>
- Aplikacja może nie znać danego FPI

XML 3 – deklaracje elementów

- Każdy element używany w dokumencie musi być zadeklarowany w DTD
- Postać deklaracji:
  <!ELEMENT nazwa zawartość >
  - nazwa to dowolna nazwa XML
  - zawartość określa jakie elementy może zawierać definiowany element i w jakiej kolejności powinny wystąpić
    - EMPTY – element pusty
    - ANY – dowolna zawartość (ale zdefiniowana)
    - (#PCDATA) – dane znakowe, treść
    - element lub elementy potomne
    - zawartość mieszana (elementy + #PCDATA)

XML 3 – definiowanie zawartości w deklaracjach

- Pojedynczy element potomny:
  <!ELEMENT kot (nazwa) >
- Sekwencje (kilka elementów potomnych w określonej kolejności), nazwy rozdzielone przecinkami:
  <!ELEMENT nazwa (imie, nazwisko) >
- Przyrostki w sekwencjach określają liczność elementów:
  - ? - element opcjonalny (zero lub jedno wystąpienie)
  - * - zero lub więcej wystąpień
  - + - przynajmniej jedno wystąpienie
    <!ELEMENT nazwa (imie?, drugie_imie*, nazwisko, panienskie+) >
- Wybory (alternatywa, element ma jeden z wymienionych elementów potomnych rozdzielonych pionową kreską):
  <!ELEMENT klient (imie_nazwisko|nazwa_firmy) >

XML 3 – definiowanie zawartości w deklaracjach

- Używając nawiasów możemy tworzyć dowolnie złożone kontrukcje:
  <!ELEMENT nazwa (imie | (imie, drugie_imie+, nazwisko)) >
- Zawartość mieszana (model opisujący element zawierający zarówno dane znakowe jak i elementy potomne):
  <!ELEMENT opis (#PCDATA|dane|inne)* >
- Uwaga: powyższy zapis jest faktycznie jedyną dozwoloną postacią deklaracji zawartości mieszanej
  - musi być *
  - #PCDATA musi być jako pierwszy element

XML 3 – deklaracje elementów – problemy

- Brak możliwości zadeklarowania *podzbioru* elementów, np. konieczność umieszczenia dwóch dokumentów tożsamości – konieczne jest zadeklarowanie osobno każdej dwuelementowej kombinacji:
  <!ELEMENT tożsamość ((dowod,prawo_jazdy) | (dowod,paszport) | (prawo_jazdy,paszport))
- Tak jest źle:
  <!ELEMENT tożsamość ((dowod|prawo_jazdy|paszport), (dowod|prawo_jazdy|paszport))
  ponieważ dopuszcza pary typu (dowod, dowod)
- Gdyby należało zapewnić dowolność kolejności to długość pierwszej deklaracji wzrosła by jeszcze bardziej

XML 3 – deklaracje atrybutów

- Prawidłowy dokument musi zadeklarować nie tylko wszystkie elementy ale i wszystkie atrybuty
- Deklaracja atrybutu:
  <!ATTLIST nazwa_elementu nazwa_atrybutu typ_atrybutu obowiązkowość>
  - nazwa_elementu określa którego elementu atrybuty opisujemy
  - nazwa_atrybutu jest dowolną nazwą atrybutu
  - typ_atrybutu – rodzaj danych jakie może przyjmować wartość atrybutu
  - obowiązkowość – czy atrybut musi występować w elemencie

XML 3 – deklaracje atrybutów - przykład

- Przykład deklaracji elementu i atrybutów:
  <!ELEMENT opis (#PCDATA) >
  <!ATTLIST opis id ID #REQUIRED autor CDATA #IMPLIED>
- Powyższy przykład deklaruje element opis o zawartości tekstowej, posiadający 2 atrybuty – id typu ID (unikalny identyfikator) oraz autor typu tekstowego.

XML 3 – deklaracje atrybutów - typy

- CDATA – tekst
- ID – unikalna wartość w ramach całego dokumentu, element może posiadać tylko jeden atrybut typu ID
- IDREF – odnośnik do identyfikatora (referencja)
- IDREFS – zbiór referencji (oddzielony spacjami)
- NMTOKEN – pojedyncze słowo (znaki takie jak dla nazw XML)
- NMTOKENS – zbiór słów
- ENTITY – encja
- ENTITIES – zbiór encji
- wyliczenie – lista stałych wartości, np.:
  <!ATTLIST doba pora (dzień|noc) >

XML 3 – deklaracje atrybutów - obowiązkowość

- #IMPLIED – atrybut opcjonalny (domyślnie)
- #REQUIRED – atrybut wymagany
- #FIXED “wartość” – atrybut stały, predefiniowany

```
<!ATTLIST dokument
        editor CDATA #IMPLIED
        mime CDATA #FIXED "text/plain"
        syntaxVersion CDATA #REQUIRED >
```

XML 3 – deklaracje encji

- XML oferuje 5 standardowych encji (< > " ' &)
- Można zdefiniować dodatkowe encje w DTD, np.
  <!ENTITY copy "©">
  <!ENTITY copyright “Copyright © 2007 xyz”>
- Odwołujemy się do nich jak do standardowych - &copyright;

XML 3 – Encje parametryczne

- Stosuje się do uproszczenia zapisów w DTD, oznaczane są dodatkowym symbolem %, np.
  Zamiast

  <!ELEMENT mieszkanie (adres, metraż, pokoje, czynsz) >
  <!ELEMENT dom (adres, metraż, pokoje, czynsz) >

  Możemy napisać

  <!ENTITY % elementy_mieszkaniowe “adres, metraż, pokoje, czynsz”>
  <!ELEMENT mieszkanie (%elementy_mieszkaniowe;) >
  <!ELEMENT dom (%elementy_mieszkaniowe;) >

XML 3 – Atrybuty czy Podelementy?

- Atrybuty
  - cechy całego elementu
  - informacje dodatkowe, metainformacje (jednostka miary, waluta, język)
  - cechy specjalne (niepowtarzalność – ID, referencje – IDREF itd.)
- Podelementy
  - obiekty podrzędne wobec elementu
  - informacja separowalna jako spójny obiekt
  - informacja powtarzalna (kilka wystąpień)
  - istotna jest koleność wystąpienia

XML 3 – Atrybuty czy Podelementy?

- Zalety atrybutów
  - zwięzły zapis
  - mogą być dodawane na zasadzie wartości domyślnych
- Zalety elementów
  - mogą zawierać inne elementy i atrybuty
  - mogą się powtarzać
  - można bezpiecznie rozszerzać ich model zawartości (dodając elemnty i atrybuty)
  - można sterować sposobem występowania (kolejnością i ilością)

XML 3 – ograniczenia DTD

- CDATA oznacza całkowitą dowolność treści
- brak wartości domyślnych dla elementów
- brak ograniczeń na unikalność wartości
- praktycznie brak kontroli typów
- brak ustalenia kolejności atrybutów
- brak wsparcia dla przestrzeni nazw
- składnia nie-XML-owa