

# **XML – arhitektura za dizajn, pisanje i upravljanje edukativnim sadržajem**

**Deo 1: Uvod u XML**

**Deo 2: DITA (Darwin Information Typing Architecture)**

**Deo 3: DITA Learning specijalizacija**

**Deo 4: Demo – aplikacija za pripremu sadržaja kurseva**

# DEO 1

## Uvod u XML

# Jezik za markiranje teksta (Markup Language)

## Takav jezik mora da definiše

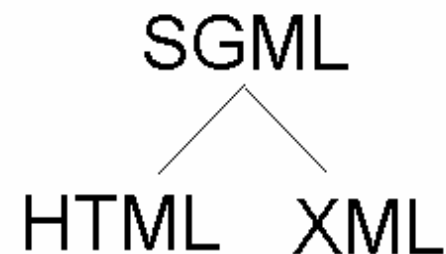
- n Koje su vrste markiranja dozvoljene
  - n Koje su vrste markiranja neophodne
  - n Kako da se markiranje razlikuje od običnog teksta
  - n Šta markiranje označava?
- 
- n **SGML (Standard Generalized Markup Language)**
  - n **Međunarodni standard za definisanje strukture i sadržaja u tekstualnim dokumentima**

# HTML (RFC 1866)

- n **HyperText Markup Language**
- n **Mala SGML aplikacija za korišćenje na web-u**
- n **Koristi samo unapred definisan skup oznaka (elemenata)**

# Šta je XML?

- n eXtensible Markup Language
- n Pojednostavljena verzija SGML-a
- n Zadržani su najbolji delovi SGML-a.
- n Dizajniran da se SGML koristi i na Web-u.
- n Fleksibilniji i prilagodljiviji od HTML-a.
- n XHTML -- prilagođen HTML 4 u okviru XML-a 1.0



# Osobine XML-a

- **Sintaksa za “označavanje” tekstualnih podataka (reči, fraze, brojevi, ...)**
- **Sintaksa je isključivo u tekstualnom formatu. XML je pisan korišćenjem *Unicode* karaktera čitljivim za čoveka (ne postoje čisti binarni podaci, tako da nema problema sa kodiranjem karaktera)**
- **Lako proširljiv. XML dozvoljava definisanje novih elementa (tipova podataka)**
- **Univerzalni format. Sintaksna pravila obezbeđuju da svi programi za obradu XML-a, MORAJU da na identičan način tumače dat XML podatak.**

# Šta XML nije?

- XML je jezik za markiranje teksta i ništa više od toga!
- XML nije programski jezik, iako se može koristiti kao osnova za čuvanje instrukcija nekog jezika.
- XML nije protokol za prenos podataka na mreži, kao što to nije ni HTML.
- XML nije baza podataka. Može se kombinovati kao neko polje unutar baze podataka.
- XML je samo XML dokument i sam po sebi ne radi ništa. Potreban je program koji će na osnovu XML-a preuzimati odgovarajuće akcije

## Razlika između XML-a i HTML-a.

**XML treba da opiše podatke, a ne da se brine o njihovom prikazu.**

n XML nije zamena za HTML.

n Različiti ciljevi:

XML je dizajniran da *opisuje* podatke i da vodi računa šta svaki podatak znači.

HTML je dizajniran za *prikazivanje* podataka i da vodi računa kako će podaci izgledati.

n HTML se brine za prikaz podataka, dok je XML zadužen za njihov opis.

# Primer XML-a

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<authors>
  <author id="3456">
    <name id="3456">
      <firstname>Dusan</firstname>
      <lastname>Kostic</lastname>
    </name>
    <bookswriten>
      <book>XML and JAVA</book>
      <book>Learning XSLT</book>
    </bookswriten>
  </author>
  <author id="7893">
    <name>
      <firstname>George</firstname>
      <lastname>Douglas</lastname>
    </name>
    <bookswriten>
      <book>Creatures</book>
    </bookswriten>
  </author>
</authors>
```

# Zašto je XML važan?

## n Čist tekst

- n Jednostavan za unos
- n Najkorisniji za smeštanje malih količina podataka
- n Moguće je efikasno smestiti i ogromne količine XML podataka koričenjem nekog XML programa u bazu podataka.

## n Laka identifikacija podataka

- n Objašnjava kakve tipove podataka XML sadrži.
- n Može se koristiti na različite načine od strane različitih aplikacija.

# Gradive jedinice XML-a -- Prolog

```
<?xml version="1.0" encoding="UTF-8">
```

n Deo XML dokumenta koji predhodni svim XML podacima

n Prolog sadrži

- § Verziju XML-a (obavezna)
  - § encoding označava skup karaktera koji će se koristiti (obično je UTF-8)
  - § Opcioni DTD (Document Type Definition)
- ```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0  
Transitional//EN" "http://www.w3.org/TR/REC-  
html40/loose.dtd">
```

# Gradive jedinice XML-a

## n Element

- § Sastoji se od otvorenog i zatvorenog taga
- § Ime elementa je ograničeno uglastim zagradama unutar tagova
- § Opšti format: `<element> ... </element>`
- § Prazan element: `<empty-Element/>`
- § Elementi se mogu ugnježditi (relacija roditelj-dete) i tako obrazovati hijerarhiju elemenata.

## n Atributi

- § Par „Ime-vrednost“ koji se javljaju unutar početnog taga elementa posle imena: `<element attribute="value">`
- § Dodatno opisuju elemente
- § Obično omogućuju podatke koji nisu deo podataka
- § Vrednost atributa mora biti pod zagradama
- § Da li treba koristiti element ili atribut?
- § Metapodaci (podaci o podacima) je poželjno čuvati kao atributi dok se sami podaci trebaju čuvati kao elementi.

# XML sintaksa

- n Svi XML elementi moraju imati završni tag, osim ako je element u praznom formatu
- n XML tagovi vode računa o velikim i malim slovima
- n Svi XML elementi se moraju uredno ugnježditi
- n Svi XML dokumenti moraju da imaju koreniti (root) element
- n Vrednost atributa mora uvek biti pod zagradama.
- n Sa XML-om, prazni karakteri se čuvaju
- n Sa XML-om, nov red se uvek pamti kao LF (line-feed)
- n Komentari u XML-u: `<!-- Ovo je komentar -->`

# XML provera

- n **Za XML dokument se kaže da je "Well Formed,, ako je očuvana XML sintaksa**
- n **Za XML se kaže da je „Validan,, ako je**
  - n XML Dokument “well formed”
  - n Očuvana sintaksa definisana unutar DTD-a (Document Type Definition) ili XML šeme
- n **XML DTD**
  - n Definiše dozvoljene gradivne blokove unutar XML dokumenta
  - n Može biti definisana u samom XML-u ili kao eksterna referenca
- n **XML Šema (XSD)**
  - n XML alternativa DTD-u, koja je mnogo moćnija
  - n Podržava namespace i tipove podataka
- n **XML sa unapred definisanim podskupom tagova ili atributa (bilo putem DTD ili XSD-a), naziva se XML Aplikacija**

## Primer DTD-a (kao deo XML dokumenta)



```
<!DOCTYPE transfers [
  <!ELEMENT transfers (fundsTransfer)+ >
  <!ELEMENT fundsTransfer (from, to) >
  <!ATTLIST fundsTransfer
    date CDATA #REQUIRED>
  <!ELEMENT from (amount, transitID?, accountID,
    acknowledgeReceipt ) >
  <!ATTLIST from
    type (intrabank|internal|other) #REQUIRED>
  <!ELEMENT amount (#PCDATA) >
  . . . . .
  <!ELEMENT to EMPTY >
  <!ATTLIST to
    account CDATA #REQUIRED>
]>
<transfers>
  <fundsTransfer date="20010923T12:34:34Z">
    . . . . .
```

# XML Šema (XML Schema)

- Nova specifikacija za specificiranje validacionih pravila za XML
- Koristi čist *XML* (nema specijalne DTD gramatike) za opis validacionih pravila.
- Šeme su daleko moćnije od DTD-a – mogu se definisati stvari kao što su celobrojni tipovi podataka, datumi, realni brojevi u određenom opsegu...
- Koriste se za validaciju tipova podataka ili za upoređivanje šema u bazama podataka sa XML modelima.

## Verzija XML Šeme za predhodni DTD primer

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="accountID" type="xs:string"/>
  <xs:element name="acknowledgeReceipt" type="xs:string"/>
  <xs:complexType name="amountType">
    <xs:simpleContent>
      <xs:restriction base="xs:string">
        <xs:attribute name="currency" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="USD"/>
              . . . . .
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:restriction>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="fromType">
    <xs:sequence>
      <xs:element name="amount" type="amountType"/>
      <xs:element ref="transitID" minOccurs="0"/>
      <xs:element ref="accountID"/>
      <xs:element ref="acknowledgeReceipt"/>
    </xs:sequence>
    . . .
```

# Klasifikacija XML dokumenta

## n **Slogoviti (record-like) XML dokumenti**

- n Unutar jednog elementa se može naći isključivo podatak ili više elemenata. Ne dozvoljavaju „mešani“ sadržaj.
- n Orijehtisani uglavnom ka podacima i dosta su restriktivniji.
- n Obično su namenjeni programima a ne ljudima
- n Primeri: konfiguracioni fajlovi (Eclipse, J2EE), XML baze

## n **Narativni (narrative-like) XML dokumenti**

- n Nema pravila šta se može naći unutar elementa za razliku od slogovitih dokumenta.
- n Orijehtisani uglavnom za opis dokumenata.
- n Namenjeni su ljudima a ne programima.
- n Primeri: HTML, DocBook, DITA...

## Primer narativnog XML dokumenta

```
<?xml version="1.0" encoding="UTF-8"?>
  <book xml:id="simple_book" xmlns="http://docbook.org/ns/docbook"
version="5.0">
  <title>Very simple book</title>
  <chapter xml:id="chapter_1">
    <title>Chapter 1</title>
    <para>Hello world!</para>
    <para>I hope that your day is proceeding
<emphasis>splendidly</emphasis>!</para>
  </chapter>
  <chapter xml:id="chapter_2">
    <title>Chapter 2</title>
    <para>Hello <emphasis>again<-emphasis>, world!</para>
  </chapter>
</book>
```

## Primer slogovitog XML dokumenta

```
<author id="3456">
  <name id="3456">
    <firstname>Dusan</firstname>
    <lastname>Kostic</lastname>
  </name>
  <bookswriten>
    <book>XML and JAVA</book>
    <book>Learning XSLT</book>
  </bookswriten>
</author>
```

## Prikaz XML-a

- n XML dokumenti ne sadrže nikakvu informaciju o tome kako treba predstaviti podatke. Predstavljanje podataka ne mora biti monitor računara već može biti knjiga ili sasvim drugi medijum.
  
- n Informacije o prikazu XML-a se mogu dodati korišćenjem
  - n CSS (Cascading Style Sheets)
  - n XSL (eXtensible Stylesheet Language)

# XSLT

- n **XML Stylesheet Language Transformation**
- n **XSLT je nastao kao deo tehnologije za prikaz tabela XSL, koji se sastoji od 3 tehnologija: XSLT, XPath i XSL-FO.**
- n **XSLT se sada koristi kao samostalna tehnologija zajedno sa Xpath-om (bez Xpatha nema rada sa XSLT-om)**
- n **Danas kada se kaže XSL, misli se na XSLT ustvari.**
  
- n **XSLT se za transformaciju (procesuiranje) XML dokumenta u neki drugi XML dokument ili čist tekst.**
- n **Najčešće se koristi za konverziju nekog XML dokumenta u HTML dokument koji se kasnije može kombinovati sa raznim CSS dokumentima za prikaz.**
- n **XSLT je nedostajajuća karika u procesu razdvajanja podataka od prikaza**

# Primena XML-a

- n **Razdvajanje podataka od HTML-a**
  - n Čuvanje podataka u zasebnim XML datotekama
  - n Korišćenje HTML-a za prikaz
  - n Promena u podacima neće zahtevati promenu HTML-a.
- n **XML za razmenu podataka**
  - n Tekstualni format je nezavistan i od softvera i hardvera.
  - n Razmena podataka između nekompatibilnih sistema, ako su unapred dogovoreni dozvoljeni elementi (DTD ili XSD)
  - n Mogu ih čitati različiti tipovi aplikacija
  - n Smanjena kompleksnost sistema za interpretiranje podataka i lakše je razumeti i doraditi isti.
- n **XML se može koristiti za smeštanje podataka**
  - n Čisti tekst fajl se može lako smestiti unutar datoteke ili baze
  - n Aplikacije se mogu lako povezati sa XML datotekama kao izvorima podataka.

# Osnovne XML tehnologije



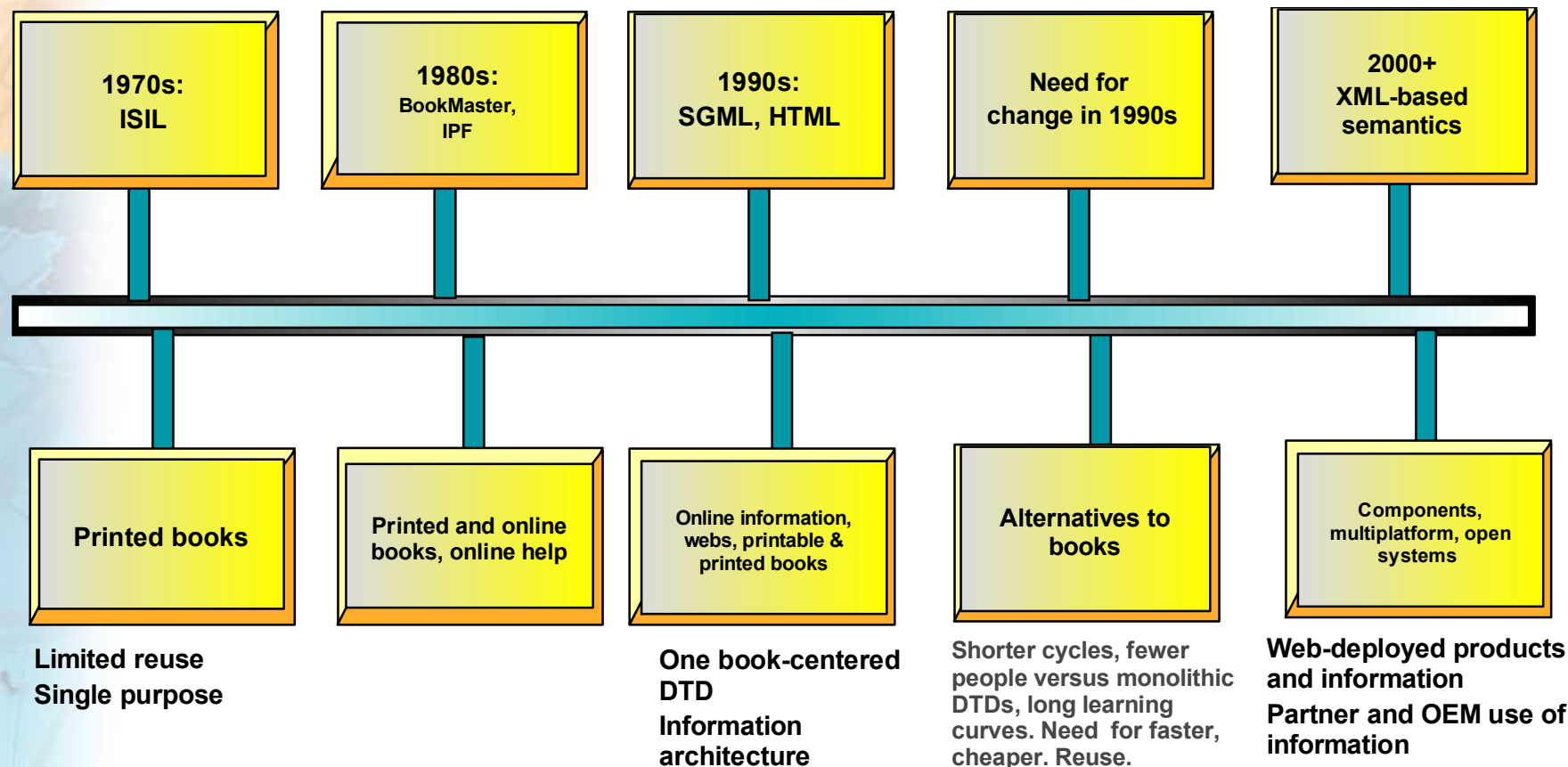
- n **DTD (Document Type Definition)**
- n **XSD (XML Schema Definition)**
- n **CSS (Cascading Style Sheets)**
- n **XPath, XPointer i XLink**
- n **XSL (eXtensible Stylesheet Language)**
- n **XSL-FO (XSL – Formating Objects)**
  
- n **Java paketi za programiranje sa XML-om**
  - n **Xerces (Java XML parser)**
  - n **Xalan (XSL transformator)**
  - n **SAX (Event-driven parsing)**
  - n **DOM, JDOM, dom4j (Document Object Model)**
  - n **JAXP (Java API for XML Processing)**
  - n **JAXB (Java API for XML Binding)**

DEO 2

# DITA

## Darwin Information Typing Architecture

# Istorija jezika za markiranje u IBM-u za potrebe tehničkih dokumentacija



## **Pronalaženje zahteva– Istorija unutar IBM-a**

- **Formiranje IBM-ovog tima za informacione arhitekture**
- **Akcent je stavljen na tehničko pisanje**
- **Razvijanje XML-a od strane World Wide Web Consortium (W3C)**
- **Napuštanje SGML ere**
- **Uočena potreba za alternativom koji bi omogućila:**
  - **Kraći razvojni tok**
  - **Različite HTML prikaze**
  - **Komponentizaciju proizvoda**
  - **Iskorišćenje komponenti koje se često koriste**

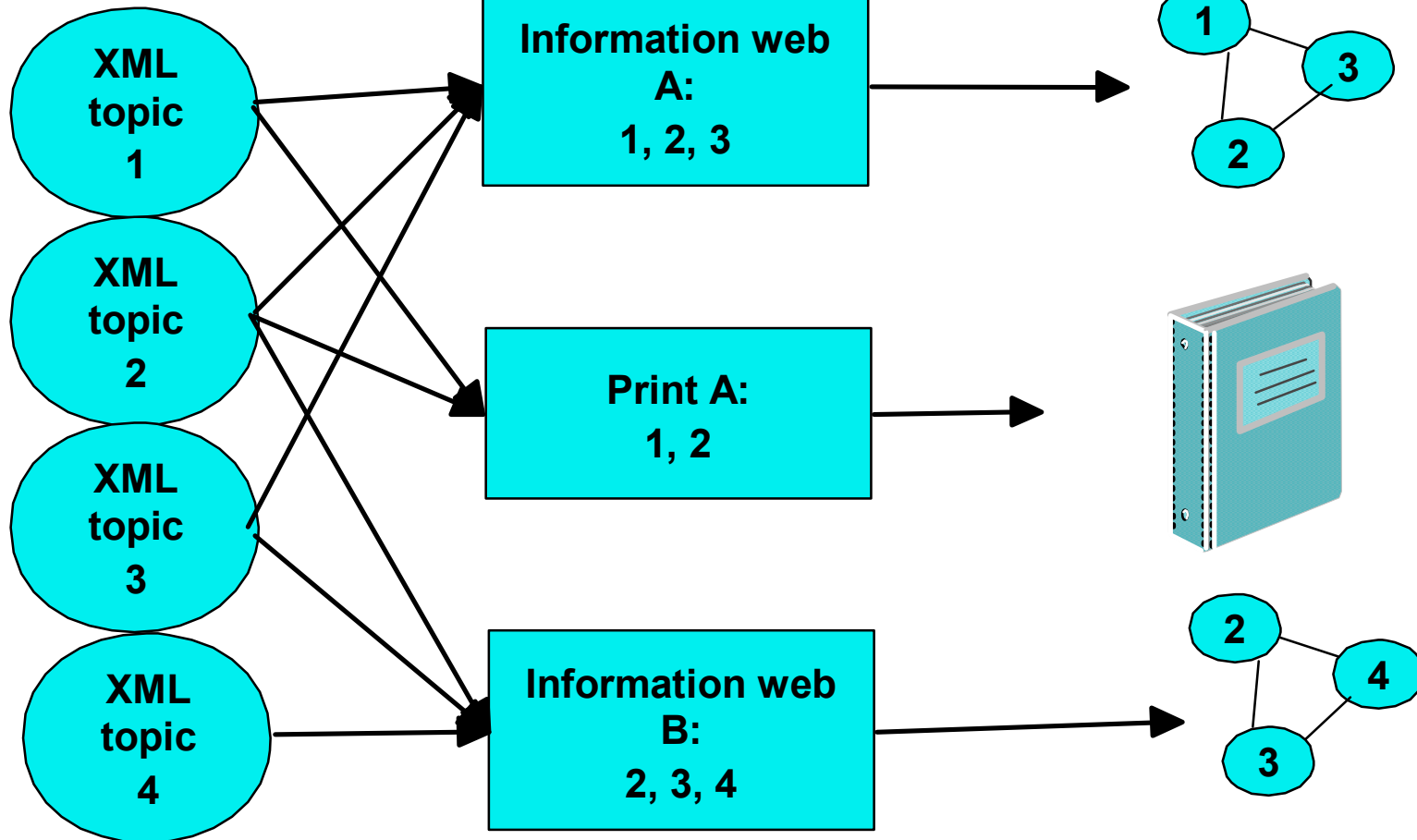
## **Pronalaženje zahteva – Potrebe kupaca**

- **Rešenje u vidu postupka, a ne konkretan proizvod za njegovo rešavanje**
- **Prezasićenost informacijama**
  - **Više smislenih informacija**
- **Zastareli podaci u knjigama**
  - **Ažuriranje i održavanje podataka**
- **Smanjiti troškove razvoja i održavanja**
  - **Omogućiti laku integraciju i održavanje informacija**

# Vizija

Single source

Multiple contexts



## Obećanja i ispunjenja o XML-u

- **Obećano**
  - **Razdvajanje sadržaja od prikaza koji bi omogućio reuse istog u ostalim prezentacionim medijima**
  - **Upotreba specifičnog markiranja za opis sadržaja**
  - **Upotreba već postojećih rešenja za laku razmenu informacija**
- **Ispunjeno – stvarno stanje običnog XML-a**
  - **Običan XML je u suštini SGML za jednostavnijom sintaksom ali i sa sličnim problemima**
  - **Postojeća rešenja možda neće biti dovoljna za svačije potrebe**
  - **Problem sa predstavljanjem znanja; nepromenljive šeme sa sobom nose već postojeće probleme kroz ceo radni vek**

## **Rešenje: DITA**

- **Prvo pojavljivanje na developerWorks marta 2001**
- **Interni protitip je napravljen za godinu dana**
- **Razvoj internih i eksternih zajednica za alate i korisnike**
- **U skladu sa otvorenim standardima i open-source-om.**

# Zašto DITA?

- **Cilj DITA je da učiti život lakšim, bar u domenu tehničkog pisanja**
  - **Za autore, za uređivače, za menadžere, za saradnike, za informacione arhitekte, za programere, za kupce**
- **DITA najviše koristi princip ponovnog korišćenja (reuse)**
  - **Akcenat je na pisanju tema (topic) na osnovu informacione arhitekture koja podržava rekombinaciju i specijalizaciju**

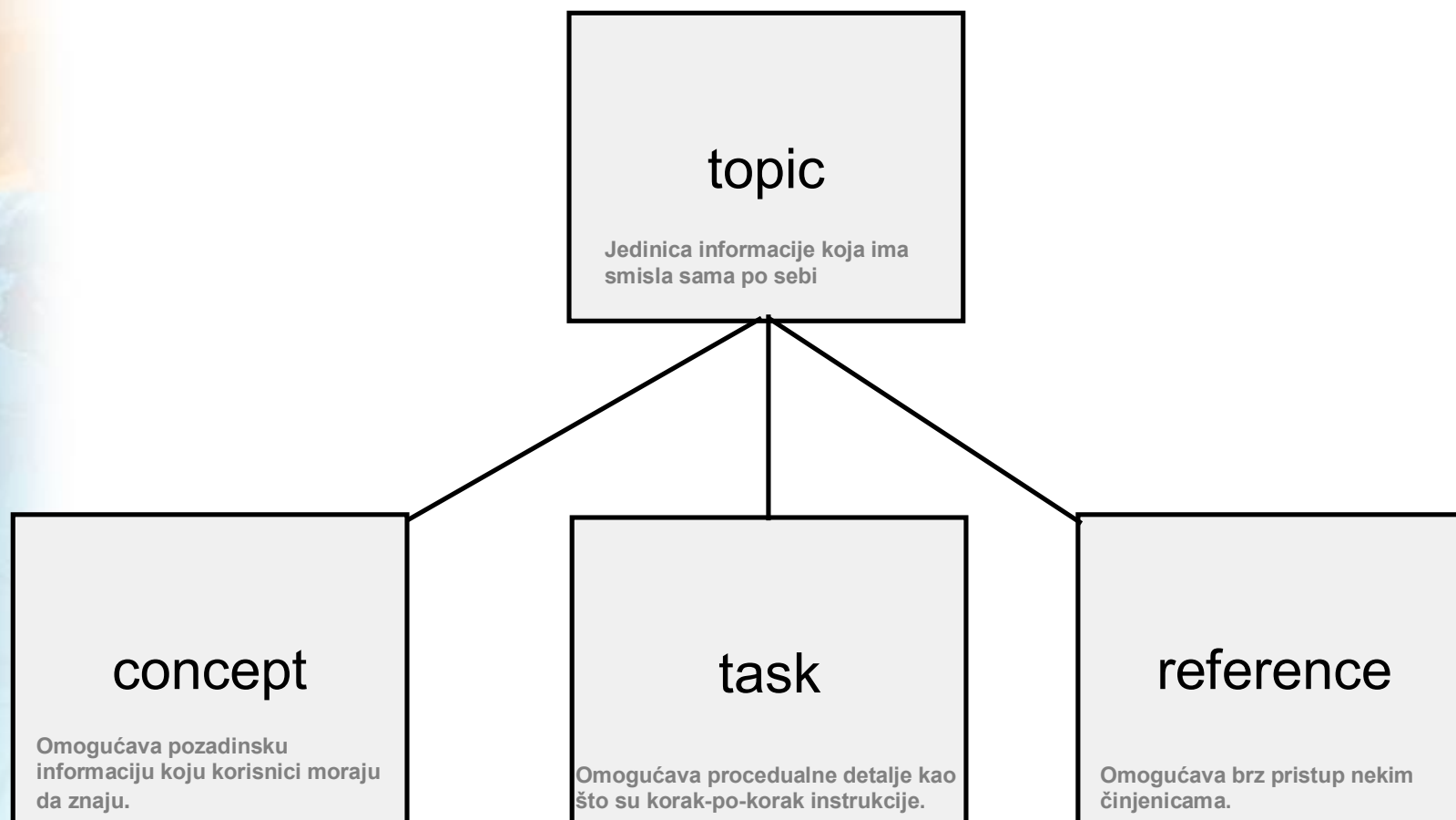
## Šta je to DITA?

- **Darwin:** DITA koristi principe nasleđivanja za specijalizacije
- **Information Typing:** DITA je originalno dizajnirana za tehničke podatke bazirane na osnovu informacionih struktura kao što su Concept, Task i Reference
- **Architecture:** DITA je model koji se može dopuniti i dizajnom i procesima

## Osnovni principi DITA standarda

- **Orijentisana ka topic-ima**
  - Diskretne (nedeljive) jedinice informacije koje pokrivaju posebnu temu u posebnom kontekstu.
- **Granulacija po topic-ima**
  - Samostalni topici se kombinuju sa ostalim topicima koji čine jedinstveni skup
- **Čvrsto tipizirana (strong-typed)**
  - DTD i šeme garantuju da DITA tipovi poštuju identične informacione strukture
- **Specijalizacija**
  - Arhitektura za proširivanje osnovnih tipova ka novim tipovima, prilagođenim za upotrebu u specijalnim sredinama i informacionim skupovima
- **Postojanje osnove bazne klase**
  - Bazna klasa je osnova za sve ostale izvedene tipove

# Osnovni DITA informacijski tipovi – Slova “IT” u DITA imenu



# DITA primer topic-a

```
<task id="installstorage">
  <title>Installing a hard drive</title>
  <shortdesc>You open the box and insert the drive.</shortdesc>
  <prolog><metadata>
    <audience type="administrator"/>
    <keywords>
      <indexterm>hard drive</indexterm>
      <indexterm>disk drive</indexterm>
    </keywords>
  </metadata></prolog>
  <taskbody>
    <steps>
      <step><cmd>Unscrew the cover.</cmd>
        <stepresult>The drive bay is exposed.</stepresult>
      </step>
      <step><cmd>Insert the drive into the drive bay.</cmd>
        <info>If you feel resistance, try another angle.</info>
      </step>
    </steps>
  </taskbody>
  <related-links>
    <link href="formatstorage.dita"/>
    <link href="installmemory.dita"/>
  </related-links>
</task>
```

Identifikacija,  
naslov i kratak  
opis

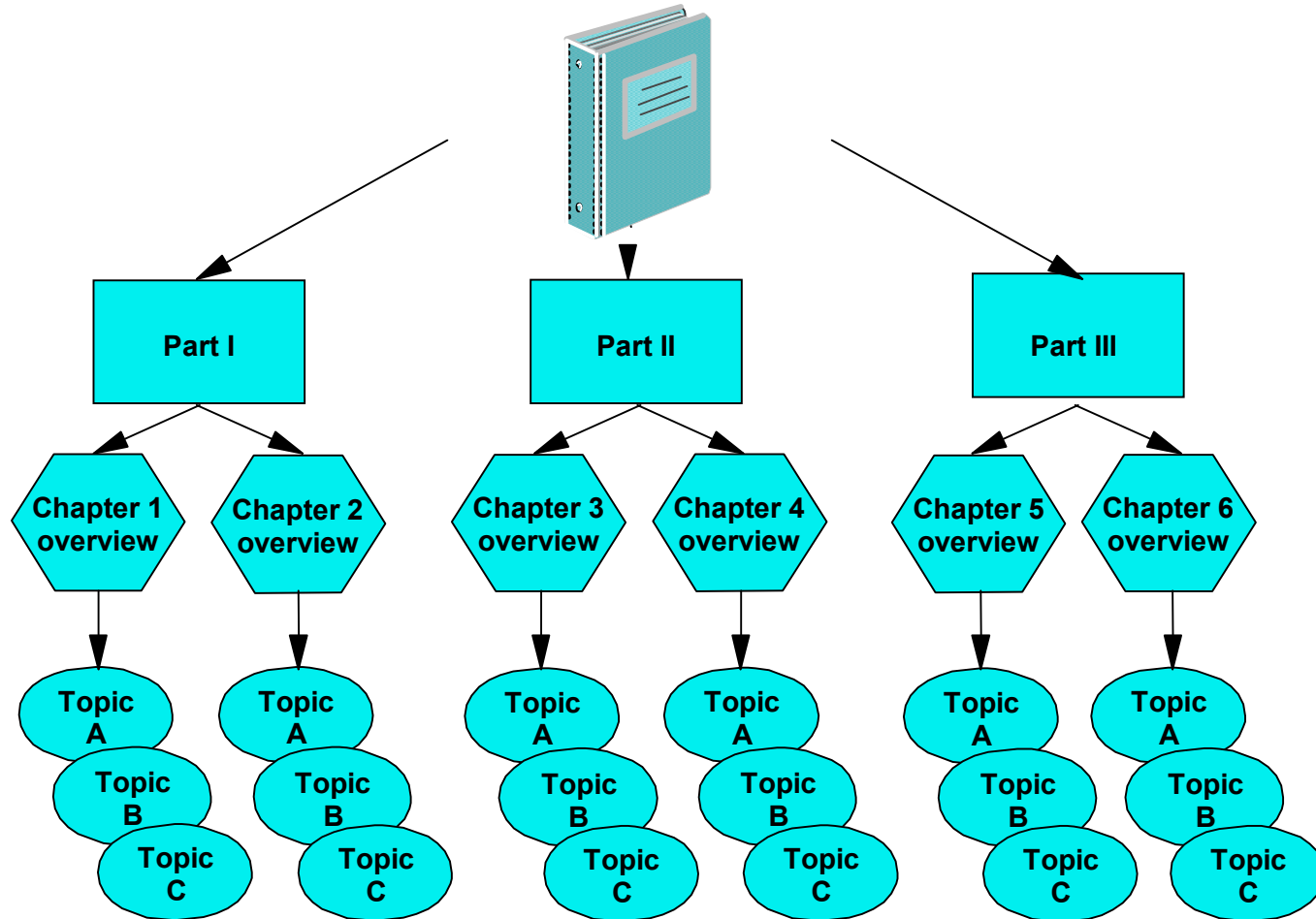
Svojstva  
Topica

Body koje je  
specifičan za  
ovu vrstu  
Topica

Veze sa  
ostalim  
Topicima

# Šta je knjiga i kako je „rasparčati“?

Knjiga je kolekcija topica (tema)



## **Krajnji prezentacioni medijum (deliverables)**

- **Krajnja DITA se na kraju može predstaviti na više prezentacionih medijuma (deliverables)**
  - HTML prezentacija
  - CHM
  - PDF
  - Help file
  - Knjiga
  - eBook za tablete
  - Audio knjiga
- **Ovo je DITA „proizvod“ koji se prosleđuje krajnjem korisniku**

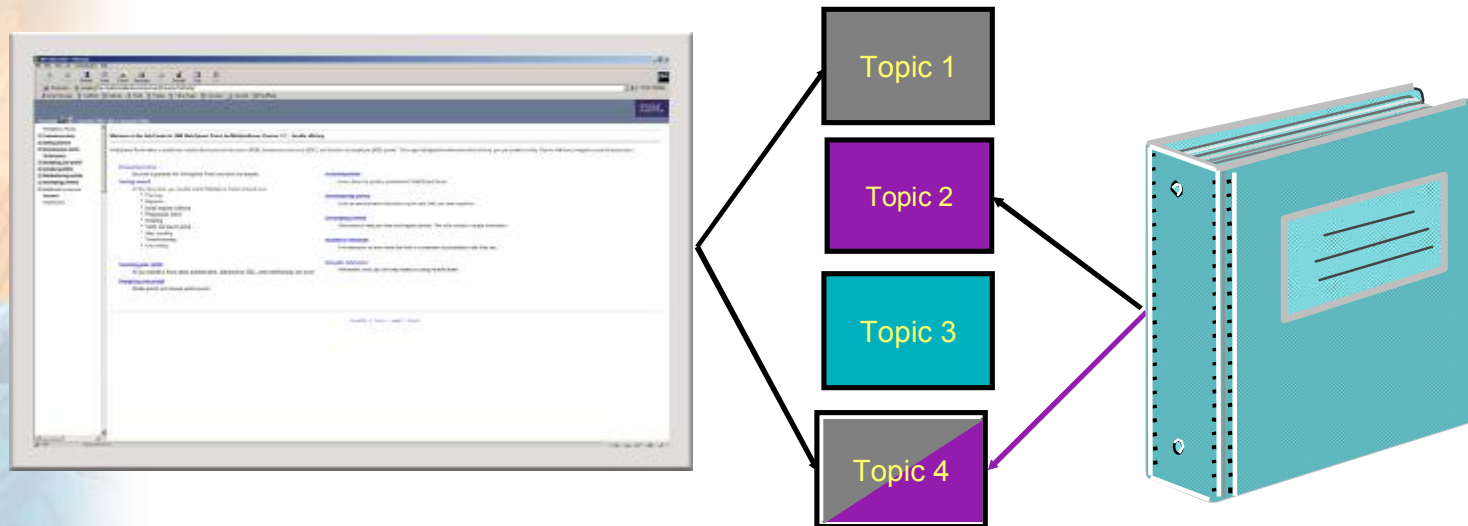
## Vrste reuse-a sa DITA-om

- **Reuse sadržaja**
- **Reuse dizajna**
- **Reuse procesuiranja za prikaz**

## **Reuse saržaja (princip granulacije topica)**

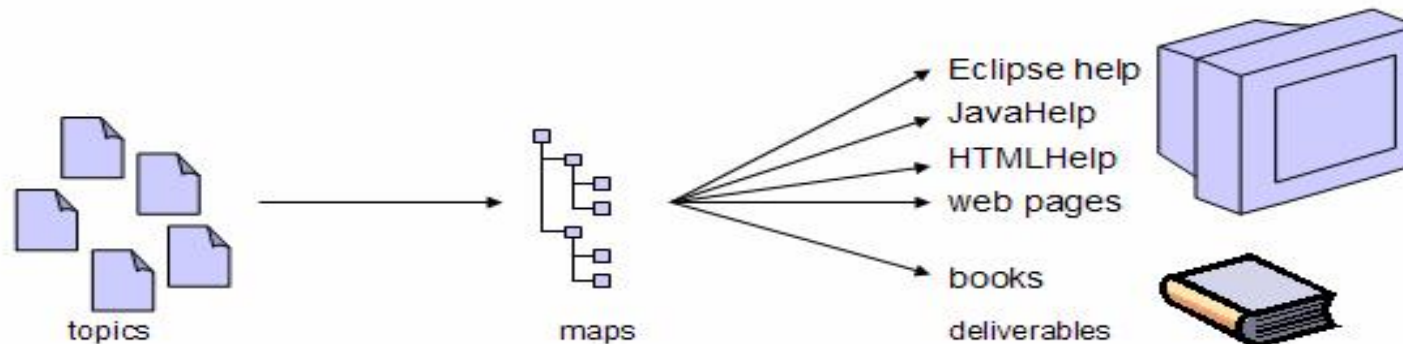
- **Reuse se jednostavno nameće tamo gde se koriste topici**
- **Ako je sadržaj pisan samostalno (da ne zavisi od ostalih topica),**
  - **Topici se mogu ponovo upotrebiti u drugačijem kontekstu**
  - **Topici iz različitih komponenti se mogu predstaviti i kao rešenje (bez potrebe pisanja novog topica)**

# Topic reuse u deliverable sistemima



- **Deliverables biraju topic**
  - Deliverable 1 koristi topic 1 i 4
  - Deliverable 2 koristi topic 2 i 4
  - Nijedan deliverable koristi topic 3

# Rad sa DITA mapama



- **DITA mapa dodaje kontekst topicima (najlakše razumeti kao TOC)**
- **Organizuje skup topica u hijerarhiju i sekvencu**  
 Različite organizacije koriste različite deliverable elemente — ne samo različite formate za isti sadržaj  
 Mogu se reusovati isti topici sa drugačijom kolekcijom topica  
 Omogućavaju višestruki pogled na iste topice: kao proizvod, kao zadatak
- **Postavlja dodatne osobine topica unutar hijerarhije**  
 Osobine uključuju ime i metapodatke  
 Promena imena na osnovu predhodnog topica  
 Metapodaci mogu da prepoznaju topic kao kompleksan za jednu vrstu deliverable-a, a kao jednostavnu za drugi

# Web-browser view / information center view

Multiple Product



Task

- § Task 1
- § Task 2
- § Task 3

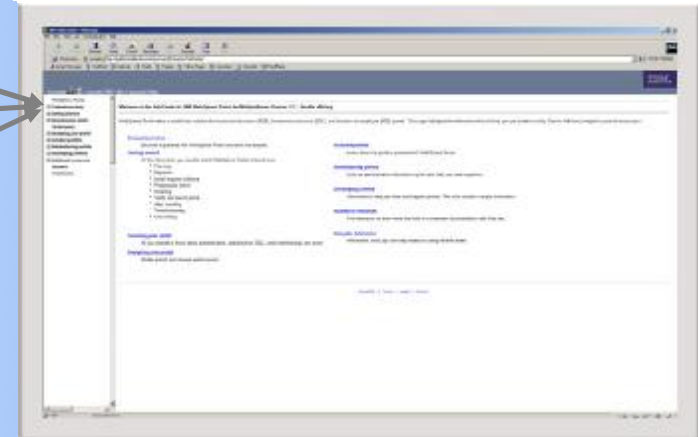
Concept

- § Concept 1
- § Concept 2
- § Concept 3

Reference

- § Reference 1
- § Reference 2
- § Reference 3

Online View



# DITA map primer

```
<map title="Tasks">
  <topichead navtitle="Installing"
    audience="admin">
    <topicmeta>
      <shortdesc>Install products before
        configuring or using them.</shortdesc>
    <topicmeta>
      <topicref href="installstorage.dita">
        <topicref href="unscrewcover.dita"/>
        <topicref href="insertdrive.dita"/>
        <topicref href="replacecover.dita"/>
      </topicref>
      <topicref href="installwebserver.dita">
        <topicref href="closeprograms.dita"/>
        <topicref href="runsetup.dita"/>
        <topicref href="restart.dita"/>
      </topicref>
      <topicref href="installdb.dita">
        <topicref href="closeprograms.dita"/>
        <topicref href="runsetup.dita"/>
        <topicref href="restart.dita"/>
      </topicref>
    </topichead>
    ...
  </map>
```

Naslov ne mora da sadrži  
topic

Ime i svojstva mogu biti  
dodeljeni u mapi

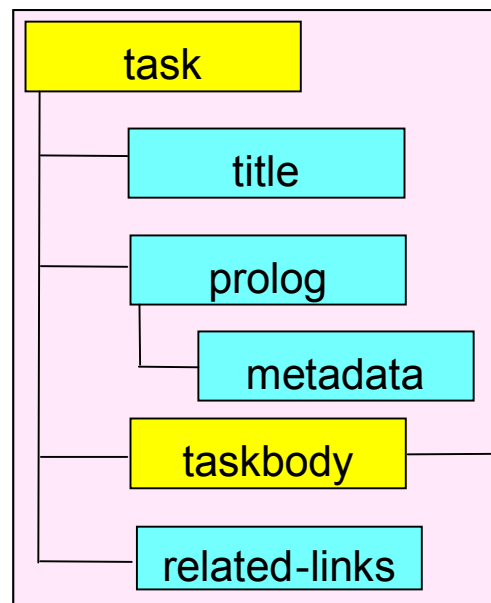
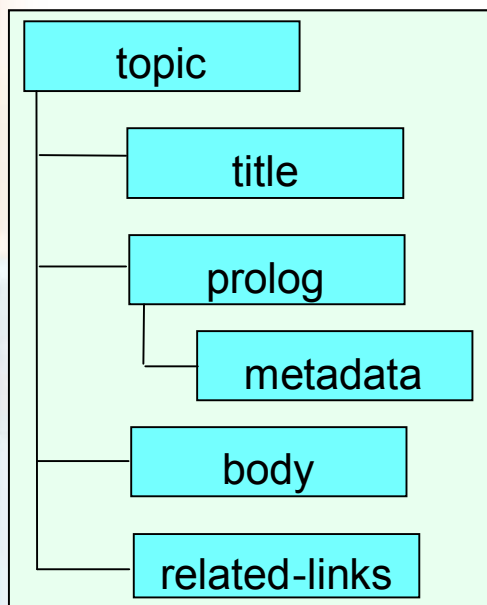
Mapa organizuje skup  
topica u hijerarhiju

Topic može da se pojavi na  
više mesta u hijerarhiji

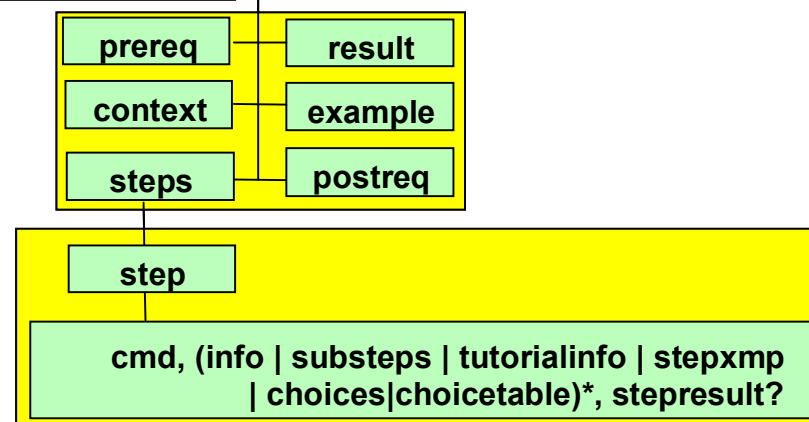
# Reuse dizajna (princip specijalizacije)

- **Opšti tipovi su retko dovoljni**
  - **Obično se zadaju uslovi koji su specifični za neku organizaciju ili industriju**
- **Dolaze zahtevi sa novim elementima**
  - **Novi element je obično specijalizacija već postojećeg elementa**
  - **Novi sadržaj je podskup već nekog osnovnog sadržaja**
- **Dodati samo izmene (delta) – još uvek koristiti osnovne elemente**
- **Dizajn je modularan**

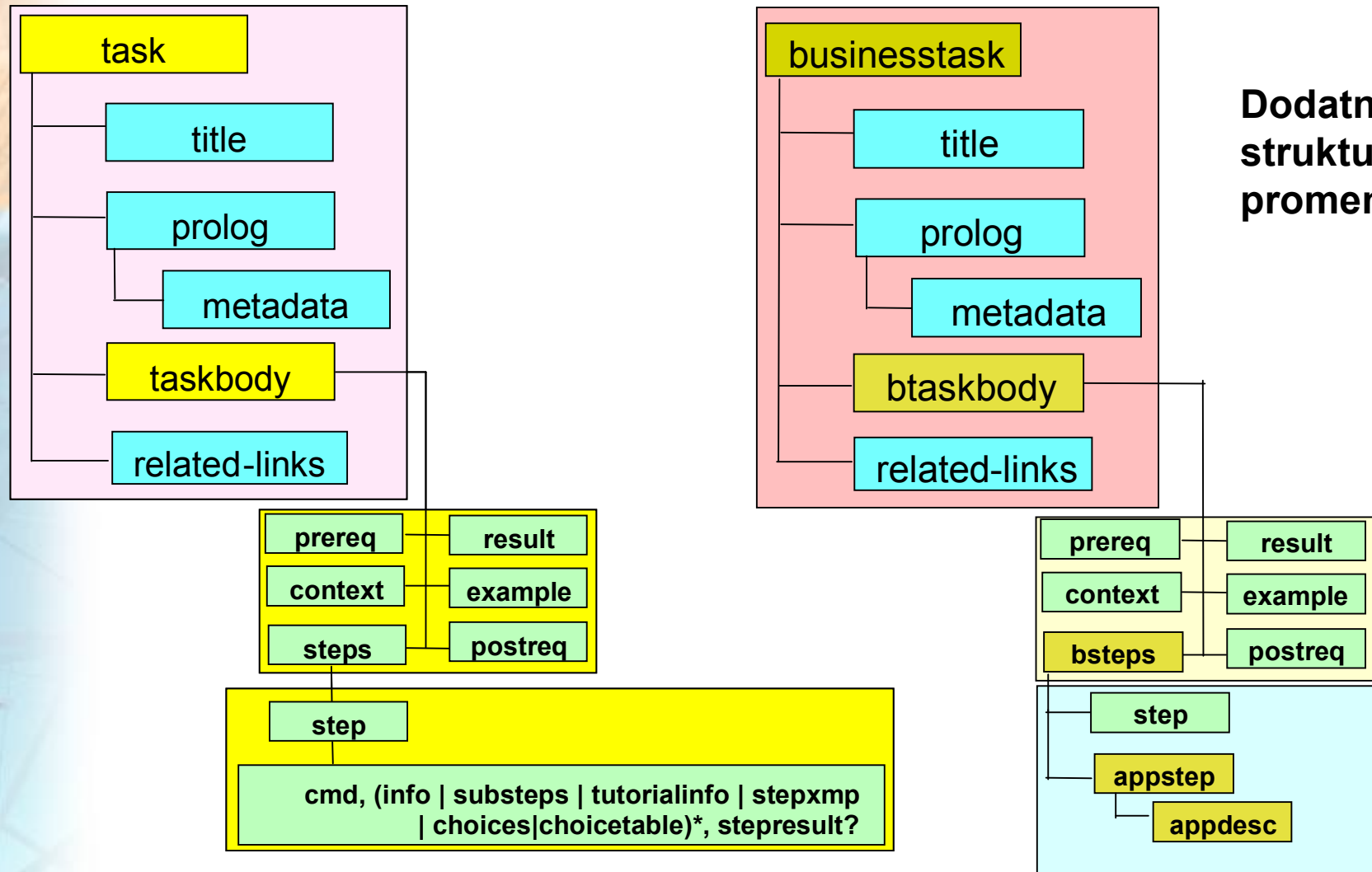
# Specijalizacija od Topic-a do Task-a



**Male dopune u DTD da bi se očuvala struktura dokumenta**

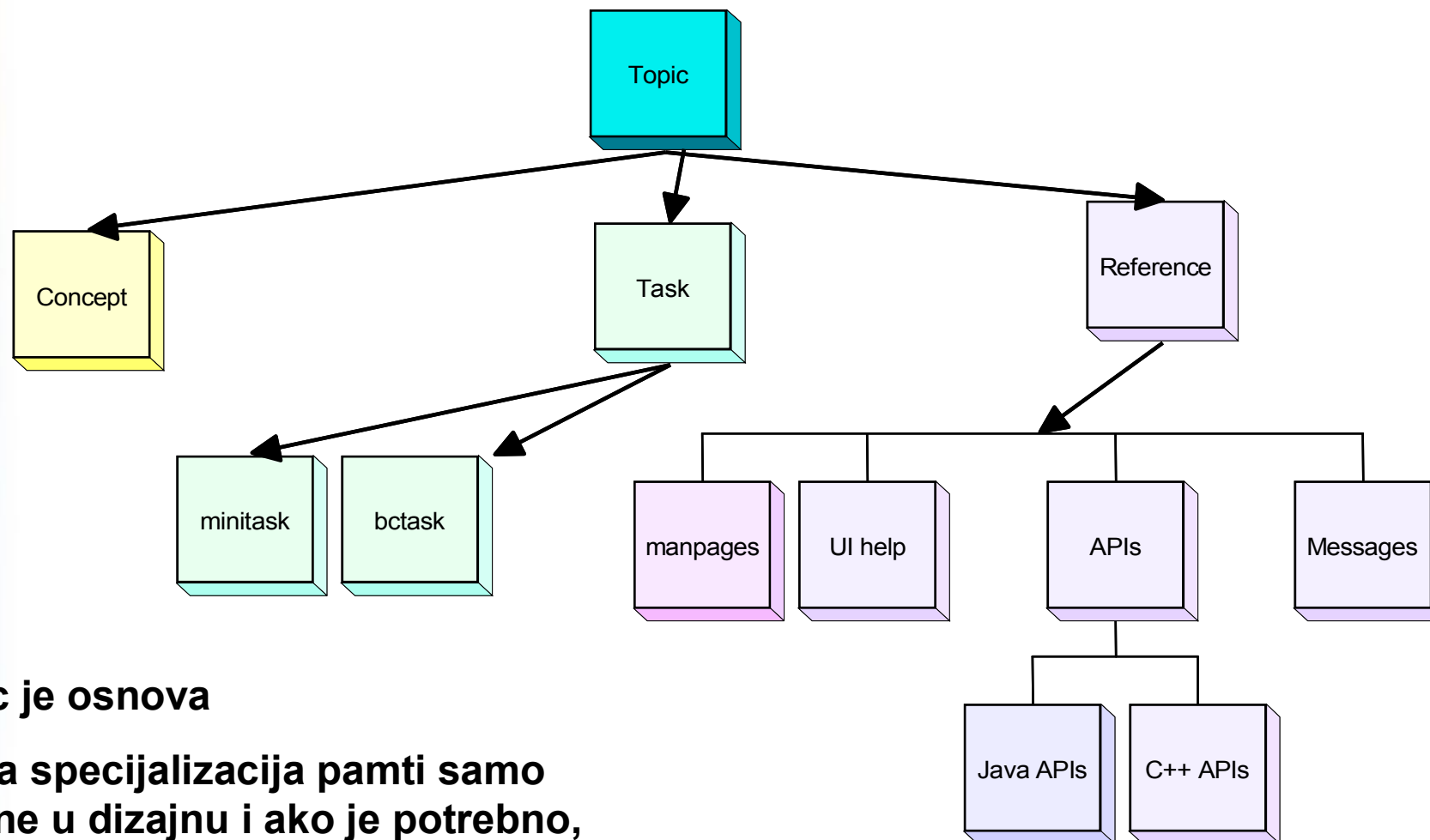


# Od Task-a do Business Task-a



**Dodatne  
strukturalne  
promene**

# Topic specijalizacije



**Topic je osnova**

**Svaka specijalizacija pamti samo izmene u dizajnu i ako je potrebno, dodatno procesuiranje za potrebe prikaza.**

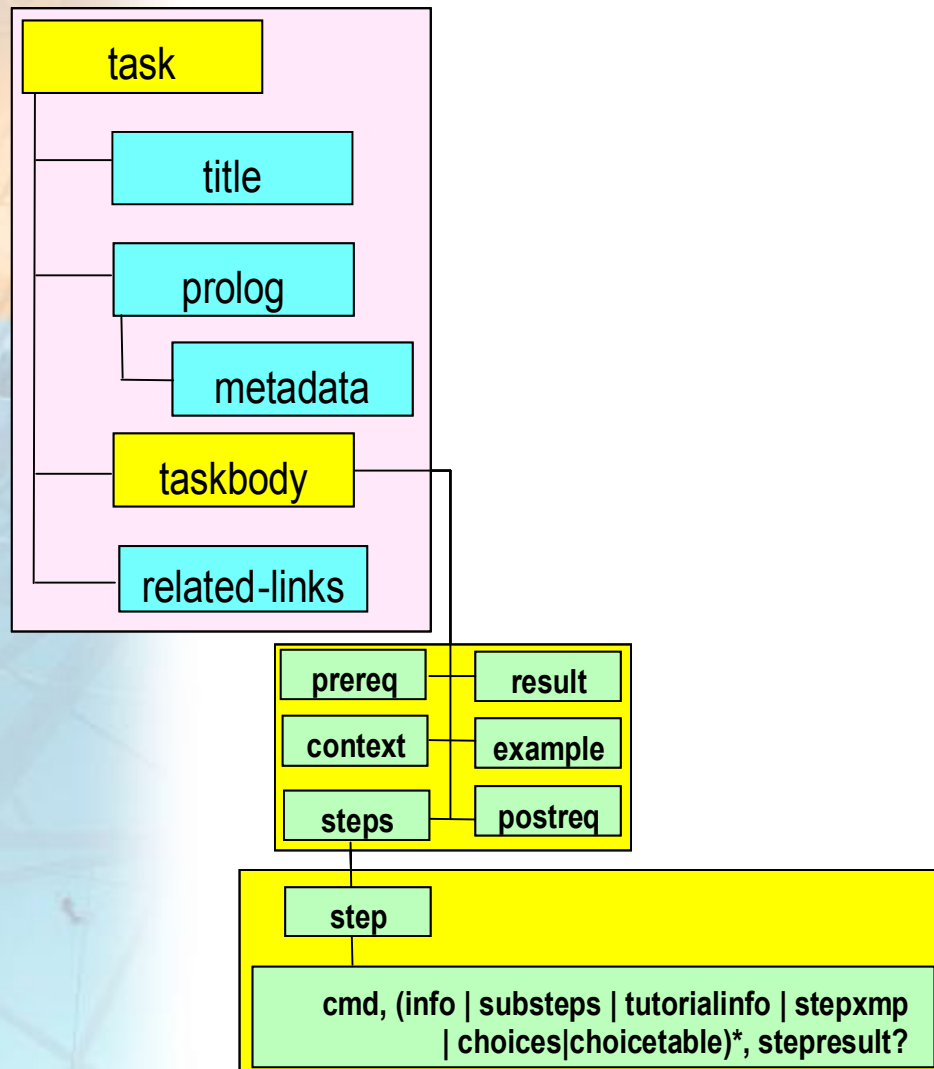
# Koristi od reuse u dizajnu korišćenjem specijalizacija

- Nema potrebe za ponovnim definisanjem osnova. Poređenja radi: kreiranje modela korišćenjem specijalizacije može trajati samo pola dana sa desetak linija koda ili pola godine sa stotinama linija koda ako se pravi od početka.
- Nema uticaja na ostale vrste dizajna koji su prilagođeni za neke druge svrhe – Izbegavanje ogromnih rečnika, jer se potrebni moduli mogu jednostavno ubaciti.
- Kompatibilnost sa osnovnim tipom – Garantovan povratak od specijalizacije do osnove
- Informacije su u skladu sa zahtevama kupaca.
- Konzistencija sa osnovnim standardima i sa zahtevima kupaca.
- Lako obučavanje novih urednika/pisaca – Umesto učenja već postojećih markera i načinima za njihovu upotrebu, pisci dobijaju samo specifične markere sa smernicama koje su već ugrađene.
- Eksplicitna podrška za različite arhitektonske zahteve – Zahtevi za razne produkte i arhitekture se mogu podržati i usloviti, umesto da se predlože i kasnije naknadno proveravaju od strane nekog supervizora.

## **Reuse procesuiranja za potrebe prikaza (princip serijalizacije)**

- **Osnova procesuiranja je u proširujućem XSLT-u.**
- **Dopuna i izmena je moguća korišćenjem nasleđivanja u procesuiranju**
  - **Procesuiranje se može prilagoditi po potrebi**
  - **Novim elementima se može dodeliti posebno ponašanje**
- **Primenjuje se procesuiranje za osnovne elemente čak i za nove specijalizovane elemente**
  - **Može se koristiti osnovno procesuiranje, ali**
  - **Može se dodati i nova/unapređena procesuiranja po potrebi**

## DITA Task



## XSLT

### Mogući izlazi

Prikazati podatke bez “koraka”, samo numerisane liste

Napraviti web stranci sa “koracima”

Napraviti PDF dokument sa “koracima”

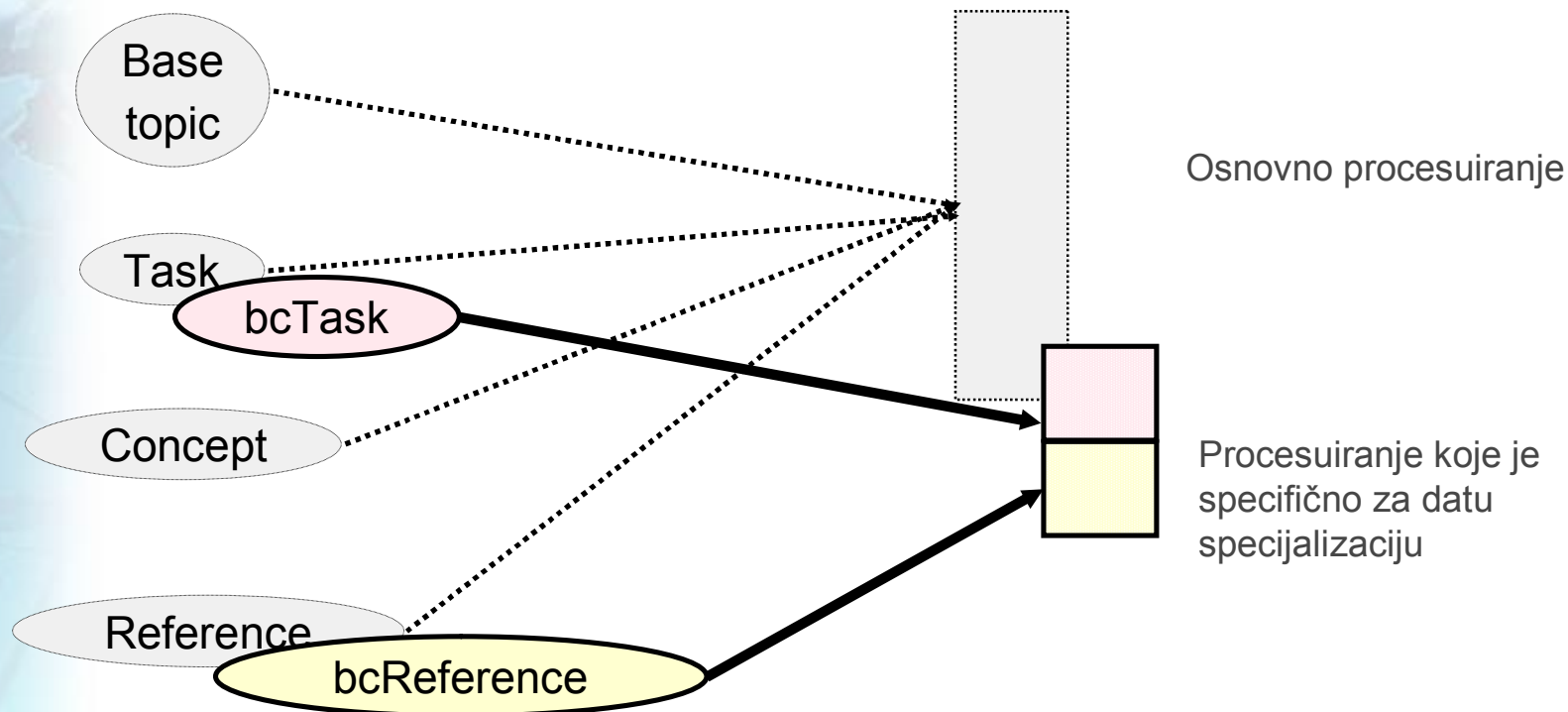
Napraviti wizard da usmerava korisnika kroz korake taska

# Specijalizacija procesuiranja

Specijalizovano procesuiranje se svodi na izvršavanje samo izmena u procesuiranju za specijalizovane topic-e

Osnovni i izmennjeni DTD

Osnovno i izmenjeno procesuiranje



## Rezime ponovnog korišćenja

- **Reuse sadržaja kroz topice**
  - Uređivanje sadržaja kao zasebna informacija
  - Reuse topica kao komponente
- **Reuse dizajna kroz specijalizacije**
  - Zadržava informacije koje su specifične za dato okruženje
  - Omogućiti kompatibilnost sa ostalim osnovnim tipovima
- **Reuse procesuiranja**
  - Nasleđivanje osnovnih procesuiranja
  - Omogućava prilagođavanje novih specijalizacija po potrebi

## **Prednosti i problemi koje je Dita rešila**

- ü Razvojni uslovi su ispunjeni**
  - Kraći ciklus razvoja**
  - Sređena varijabilnost u HTML prikazima**
  - Podržana komponentizacija i potreba za ponovnim korišćenjem**
- ü Pružila je rešenje postupka, a ne krajnji proizvod**
- ü Rešen problem prezasićenosti informacijama**
  - Više smislenih informacija**
- ü Zastarele podatke u knjigama**
  - Ažuriranje i održavanje informacija**
- ü Smanjivanje troškova održavanja**
  - Prilagođavanje i ažuriranje podataka**

## DITA je Open standard

- **Dizajn je otvorenog tipa – DITA održava OASIS**
  - **Organization for the Advancement of Structured Information Standards**
  - **OASIS TC ima odlučnu podršku od strane proizvođača, saradnika, implementatora i korisnika**
    - **Sastanci su otvoreni za sve zainteresovane goste**
  - **DITA na OASIS-u**  
[http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=dita](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dita)
- **Kod je otvorenog tipa**
  - **Postoji ogromna zajednica koja pravi besplatne alate**
  - **DitaOT – DITA Open Toolkit**
    - **DitaOT će nastaviti da se vremenom usavršava, dok ne postane svemoguć proizvodni sistem. Za sada se najviše koristi za demonstraciju DITA mogućnosti**
  - **Stalno se proveravaju prioriteti i potrebe u ovoj Open Source zajednici**

DEO 3

# DITA Learning specijalizacija

# Opšti postupak prilikom pravljenja kursa

- Analiza zahteva kursa za određenu oblast
  - Koji je cilj učenja i kako ga postići?
  - Uvođenje i pregled materijala za kurs
  - Kontekst u kojem će se koristiti materijal
  - Dodavanje materijala za učenje
  - Rezime materijala za učenje
  - Dopunjavanje materijala kroz vežbe, simulacije ili zadatke
  - Ocenjivanje naučenog gradiva
  - Pregled dobijenih rezultata od ocenjivanja
- 
- Postupak nameće dve vrste informacija: sam materijal koji će se koristiti za učenje (informacione komponente) i materijal koji će opisivati postupak učenja (instrukcione komponente). Na ovaj način se promoviše instrukcioni integritet.

## Zašto koristiti DITA za učenje?

1. Problemi sa nestruktiranim podacima. Koje probleme ovo predstavlja za razvoj materijala za učenje?
2. Prednosti struktuiranog sadržaja za učenje. Šta struktura sama po sebi donosi?
3. Zašto baš DITA za učenje? Koje su njene prednosti?
4. Dita learning and training specijalizacija Koje nove elementi uvodi?

## Šta je strukturirani sadržaj?

- Strukturirani sadržaj je sadržaj koji se pridržava predefinisane organizacije. Taj sadržaj je označen ili opisan markerima (tagovima) organizovanim u hijerarhiju. Semantički tagovi se mogu napraviti da opišu sam sadržaj u fleksibilnom obliku, čineći podatke između tagova jasnim.
- Npr. Veoma je lako zaključiti svrhu i nameru sledećeg sadržaja između warning tagova:

```
<warning>Ne uključivati aparat u vlažnoj sredini.</warning>
```

## DITA learning and training specijalizacija

- DITA za učenje tehničkih specifikacija predstavlja dobar početak za one profesionalce koji traže da iskoriste prednosti DITA za potrebe učenja. Specifikacija je napisana od strane pod-komiteta unutar OASIS DITA TC, koji je sastavljen od DITA eksperata i profesionalaca na polju učenja. Svi oni imaju veliko interesovanje u pravljenju specijalizacije koja se bavi potrebama zajednice o učenju.
- Ove specifikacije ne predstavljaju završnu tačku, već predstavljaju novi početak i osnovu odakle će se DITA proširiti u svet učenja i obučavanja.

## Zašto baš DITA za ove potrebe?

- DITA sama po sebi razdvaja podatke od njihovog predstavljanja.
- DITA je otvoreni standard i samim tim je daleko pristupačniji i jeftiniji od ostalih zatvorenih sistema, koji vremenom zastarevaju.
- DITA maksimalno koristi reusing, tako da to smanjuje duplikate i olakšava razvoj materijala za učenje.

# Tipovi topic-a unutar Learning specijalizacije

- Learning plan
  - Learning plan omogućava timu dizajnera za učenje da obezbedi sve relevantne podatke o potrebama prilikom učenja i o tome kako će se pristupati njihovom rešavanju. Ovo je namenjeno profesionalcima na polju učenja da osmisle čist put procesa učenja neke oblasti pre nego što bilo kakvo uređivanje uopšte i počne. Podaci koji se nalaze u ovom dokumentu će dati smernice i pomoći svim osobama, uključene u razvoju datog kursa, da sagledaju ceo proces učenja i da se u skladu sa njim ponašaju.
- Learning overview
  - Learning overview predstavlja ciljeve prilikom učenja, uključujući ostale informacije koji će biti od pomoći korisniku, kao što su preduslovi, vreme trajanja, ciljna publika i može da sadrži informacije i strategije koji će biti od pomoći. Prosto rečeno, ovo je ptičija perspektiva celog kursa koji pomaže korisniku da zna šta da očekuje kako napreduje kroz kurs.

# Tipovi topic-a unutar Learning specijalizacije

- Learning summary
  - Learning summary rekapitulira stečeno gradivo i obezbeđuje završne smernice koji će pomoći korisniku da gradivo lakše zapamti i može sadržati i par pitanja za proveru znanja. Learning summary jednostavno sumira sve što je naučeno o datom kursu.
- Learning content
  - Learning content predstavlja materijal za učenje koji omogućuje direktnu upotrebu već postojećih DITA materijala kao što su task, concept i reference topic, kao i odgovarajući dodatni sadržaj. U Learning contentu korisnik može naći materijal koji će koristiti prilikom učenja.
- Learning assessment interakcije
  - Learning assessment predstavljaju instrumente koji ocenjuju napredak, ohrabruju preslišavanje i stimulišu dalje učenje materijala unutar kursa.

## Interakcija unutar Learning Assessment-a

- Unutar Learning Assessment-a se mogu naći pitanja na koje korisnik odgovara
- Realizacija pitanja zavisi od deliverable-a koji se koristi: HTML ima polja na koje korisnik može da odgovara dok knjiga to nema
- Postoje nekoliko vrsta pitanja:
  - Tačno/netačno, izabrati jedan tačan odgovor, izabrati više tačnih odgovora, spajanje tačnih odgovora i hotspot.

# Learning mapa umesto DITA mape

- Unutar learning specijalizacije se koristi learning mapa umesto DITA mape
- Sadrži reference na learning tipove podataka
- Reference moraju da budu u odgovarajućem redosledu
- Postoje kontejneri koji mogu da grupišu srodne learning topice u istu celinu

## DEO 4

# Demo – aplikacija za pripremu sadržaja kurseva

# Studija slučaja

- Mobilni klijent za kreiranje DITA Learning komponente
- Pretpostavke:
  - CMS Alfresco RESTfull API za login, list, download, upload
  - Mobilni uređaj Tablet ili smart phone

CILJ:

Predložiti:

Dizajn, arhitekturu i pokaznu vežbu (opciono) za klijentsku aplikaciju koja ima funkcije kreiranja, editovanja i prikaza Learning komponenti po uzoru na Z-Aphrodite web aplikaciju.

Predlog treba da sadrži izbor platforme, S/W alata za realizaciju, use case diagrame, skicu korisničkog interfejsa...