

EP zapiski

E-poslovanje – gradniki področja

- Posl. Funkcije, procesi, verige dodane vrednosti, poslovni modeli
- Tehnološki modeli in rešitve
- Komplementarne dejavnosti

Poslovne funkcije se delijo na

- Procesne (nabava, proizvodnja, prodaja)
- Prečne (raziskava in razvoj, delo s kadri, finance, trženje, logistika)

Nivoji upravljanja:

- Strateški – dolgi rok
- Taktični – udejanjanje strateškega
- Operativni – aktivnosti za realizacijo procesov, določenih v taktičnem

Informacijski sistemi proizv. Organizacij imajo 4 nivoje:

- »klasični« poslovni
- Proizvodni (razvrščanje opravil, sledenje od surovin do izdelka)
- Nadzorni (upravljanje operacij)
- Procesni (izvedba operacij)

E-poslovanje najprej v storitvenem sekotruju

Nivoji procesnih IS

- Procesni: nadzor nad osnovnimi procesi (senzotji, aktuatorji → krmilniki)
- Nadzorni: zbira podatke od in nadzira procesni nivo (SCADA – Supervisory Control And Data Acquisition)
- Proizvodni nivo: MES – Manufacturing Execution Systems sledijo surovina -> izdelek in prejemajo navodila in pošiljajo podatke poslovnemu nivoju

- Poslovni nivo: Na prvem nivoju so MRP – Material Requirements Planning, nad njimi ERP – Enterprise resource planning, vrh pa direktorski IS. Tu so strateške odločitve, na nivoju MRP pa delno že tektične.

Digitalna ekonomija (zakaj sta na slajdu 61 zamaknjeni črki?)

Sistemski način razmišljanja (Sistemska dinamika)

Dejavniki, ki povzročajo kompleksnost sistema:

- Dinamičnost
- Sklopljenost
- Nelinearnost
- Zgodovinska pogojenost
- Samoorganiziranost
- Adaptivnost
- Vztrajnost

Modeliranje in simulacije so pomembne, mentalni modeli so prešibki

- Vzročna povezava
- Povezava z zakasnitvijo (škatla na povezavi)
- Pozitivna(ojačevalna)/negativna(stabilizirajoča) zanka
- + povezava → če se poveča X se poveča Y in obratno
- - povezava → če se poveča X se zmanjša Y in obratno
- Zaloge (akumulacija)

Tipične oblike obnašanja dinamičnih kompleksnih sistemov:

- Eksponentna rast
- Stabilizacijski trend (se znižuje in konvergira proti nekemu nivoju)
- Rast tipa S -> nivo, lepo zraste, nivo
- Oscilacija – pač nihanje, tega nečeš like ever

- Rast s prenihavanjem
- Rast s sesutjem – bad

Št. 83 DVZ tipičnih oblik (kako zgledajo povezave, ki privedejo v stanja)

Ostale oblike

- Ravnotežje
- Naključno obnašanje

Veriga dodane vrednosti

Dobavitelj -> posrednik -> (pro)izvajalec -> posrednik -> kupec, nazaj gre pa informacija

Stem da je zadeva napredovala tako, da je blo najprej vse fizično, na koncu je blo pa skoz več prek interneta (pač informacije se pretakajo, in ne več samo nazaj)

Danes – more or less

Dobavitelj <-> posrednik <-> kupec, s tem da so vsi povezani prek interneta pa feedback pa to

Inovacije – neki delaš na nov način, tako da je vse skupi bolje

Za eposlovanje je tehnologija ključna, sam jo morš znat izkoristit

Strategije – bit hočeš boljši od konkurence za čim manj dnarja

Poslovni model – se neposredno veže na strategijo. Izoblikuje vrednost storitve oz proizvoda za stranko na osnovi tehnologije.

IT lahko pomaga dodajat vrednost poslovnemu modelu.

Poslovni modeli e-poslovanja izoblikovani do konca 90-ih let

- Oglaševanje
- Virtualne trgovine

- Portali

Danes:

Domena tehnike in tehnologija <-> Poslovni model (strategija, dodana vrednost, stroški vs dobiček ...) <-> širša družbena domena

Modeli odločanja

Značilnosti managementa

- Veliko raznolikih nalog (+ njihova razdrobljenost), hitro opravljanje le teh
- Ad-hoc
- Kompleksna mreža interakcij
- Močna privrženost neposredni (besedni, pogovorni) komunikaciji
- Neki z lastnim voznim redom

Nekateri osebni modeli:

- Racionalen model – cilji, možne alternative, izbira najboljpe
- Model hitre zadovoljitev – podobno kot zgori, sam alternativ je manj, izbereš prvo ustrezno
- Model z zmedenim pristopom – cilji in predvidevanje posledic, merilo je sprejemljivost s strani ostalih
- Celovit psihološki model (ljudje bolj občutljivi na negativne posledice, nekonsistenten odnos do malo verjetnih dogodkov, raje sprejmejo negativen izid, če je predstavljen kot strošek in ne kot izguba. problem mentalnih fiksacij)

(cca str 122)

Modeli odločanja v organizaciji (Laudon):

- Racionalen – najboljša izmed alternativ
- Birokratski – cilj je preživetje, cilji se določijo glede na materialna sredstva in kadre
- Političen – cilj je uresničitev politične ideologije (odločitve temeljijo na vplivu in moči)
- »smetnjak« - odločitve temeljijo na interakcijah med udeleženci, ki rešujejo začasne probleme

Volge managerjev (Mintzberg)

- Medosebna vloga
- Informacijska vloga
- Vloga odločanja

Vse to podprto z IS

Nivo/vrsta IS vs nivo odločanja



Internet je pomemben in je novo poslovno okolje. Pomembno je razumevanje tehnoloških trendov, organizacijskih sprememb in pričakovanj potrošnikov. Predvidevanje in prilagajanje ima velik pomen.

Tehnološki trendi – integracija (na polno, povsod) -> na nivoju združbe, + vse podatke v sistem, ne glede na naravo

Organizacijski trendi

- Zahteve po visoko kvalificiranem kadru
- Kapitalsko (ne?)intenzivne dejavnosti

- Ohranjanje le osrednjih poslovnih funkcij, ostalo gre v oddajo del (outsourcing)
- Vzpostavljanje večje transparentnosti poslovanja (za stranke in managerje)
- Nenehno, trajno inoviranje
- Upravljanje virtualnih organizacij

Pričakovanja strank

- Hitrost, kakovost in cena
- Samopostrežnost
- Prilagajanje proizvodov in storitev zahtevam odjemalcev(strank)
- Enotno dojemanje dobavitelja ne glede na vrsto/način komunikacije (telefon, email, ...)

Vpliv e-tehnologije – povzetek

- Tehnologija ni več sekundarni, ampak je ključni dejavnik in gonilna sila pri snovanju in izvedbi strategije
- Sposobnost strukturiranja in upravljanja pretoka informacij je močnejše in učinkovitejše orodje kot kakršnakoli druga vrsta upravljanja
- Nesposobnost zavreči dominantne in ustaljene poslovne vzorce – čest razlog za neuspeh
- Personalizacija – e-poslovanje omogoča podjetjem in ustanovam, da vzpostavijo bolj oseben način komunikacije in prisluhnejo specifičnim željam strank

- Personalizacija – stranka dobi novo vsebino in izkušnjo
- Poslovanje in informatizacija – osnovna naloga managementa postaja hipno usklajevanje poslovnih procesov z aplikacijami
- Prilagodljivost – poslovni modeli za e-poslovanje morajo biti izrazito prilagodljivi

Management = načrtovanje, organiziranje, vodenje, nadzorovanje, ukrepanje (ciklično za vse)

Nova paradigma – never plan further than 24 months out – ker se zadeve prehitro spreminjajo

BPR – business process reengineering

- Radikalne spremembe poslovni procesov
- Vsi elementi podvrženi analizi (npr. z viharjenjem možganov)
- Koristne ideje se izvedejo v okviru IS

Razvijanje celovitih e-poslovnih rešitev

- Zahtevana so znanja za integracijo (strategij, procesov, aplikacij, podporne tehnologije)
- 2 vrsti načrtovanja
 - o Analitično, od vrha navzdol
 - o Taktično – operativno, od dna na vrh

Analitično

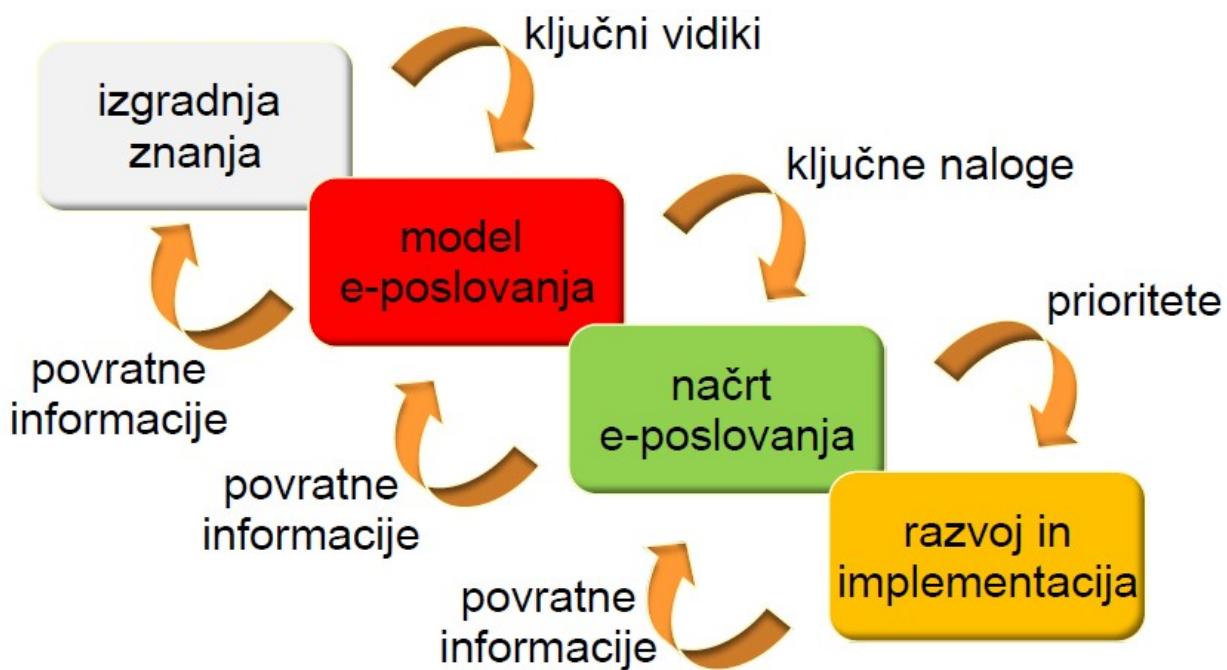
- Sistematično izdelovanje in vrednotenje možnih scenarijev v bodočnosti na podlagi obstoječih podatkov za določitev strategije
- Težava: to je primerno za stabilno okolje, poslovanje pa je v izrazito fluidnem okolju
- Druge nevarnosti:
 - o strateški načrti, ki so sami sebi namen ali brez oprijemljivih ciljev
 - o nobenih povratnih informacij

Taktično (bottom-up)

- v e-poslovanju je spremenjanje norma, zato so kratkoročne adaptacije pogoste
- tako razvite aplikacije pogosto nezdružljive z obstoječimi sistemi
- posledično je načeta celovitost in konsistentnost inf. kanalov, ki onemogoča učinkovito vodenje

Rešitev – trajno načrtovanje s povratno zvezo (TNPZ) -> zlitje prejšnjih dveh strategij

- kakšne so potrebe odjemalcev in sposobnosti za njihovo izpolnitev?
- Dorečen model eposlovanja
- Načrt eposlovanja – kaj storiti → kako storiti
- Razvoj aplikacij in izvedba z uporabo: integrirana izvedba, obveščanje vseh zaposlenih o dogajanju



TNPZ je osnova za agilne metode (AM)

- Agilni razvoj je upravljavski pogled na razvoj prog. opr.
- Primer AM je SCRUM, ki je popolno nasprotje tradicionalni metod, kot je slapovna, kjer so 3 faze
- razvoj, implementacija in vzdrževanje

Vodila za razvijanje eposlovanja

- Izgradnja znanja (zasnuj strategijo glede na podatke o strankah)
- Ozena zmožnosti
- Namenska faza - spoznaj lastno organizacijo
- Določitev modela (primeri na št 161)
- Načrt aplikacij(e), njih uvedba in izvedba
- Celovito vrednotenje novega modela

Zahteve za IS - vodila

- Razpoložljivost 27/7/365
- Odziv max 5s, nemoteče je 2s

- Enostaven vmesnik, če se da poosebljen
- »100%« pravilnost delovanja
- Sočasna podpora (velike količine uporabnikov)
- Modularna zgradba s pomočjo komponent (za spletnne rešitve)
- Integracija (najprej monolitno, danes komponentno)

Komponente

- Niso samostojni programi, ampak tečejo znotraj t.i komponentnih vsebovalnikov
- poznajo dostopne funkcije vsebovalnika, vsebovalnik pa lahko dinamično ugotovi funkcije komponente
- Delimo na strežniške in odjemalske, pa tudi trajne in začasne
- Močno olajšano nadgrajevanje programske opreme
- Komponente vs objekti - objekti naj bi bili enostavljeni, komponente pa naj bi omogočale granularne klice - več funkcij na zahtevo, ker so klici prek omrežja časovno dragi
- Ključne besede: prilagodljivost, sočasnost in redundančnost

Tehnološka vodila in standardizacija

RIP -> EDIFACT - potrebno je bilo preslikovati med predstavitvam, zaželena avtomatizacija → standardizacija

RIP je izmenjava podatkov med računalniki organizacij v obliki elektronskih poslovnih dokumentov, primernih za avtomatsko obdelavo.

RIP v ZDA določa ANSI, v Evropi pa EDIFACT

RIP zmanjša zakasnitve, napake in stroške transakcij

Gradniki dokumentov:

- El. Ovojnica
- Glava funkcionalnega sklopa podatkov
- Glava transakcije
- Glava podatkovnega segmenta
- Podatkovni element, ki je osnovna enota

Definicijska tabela sporočil:

- Kaj se lahko pošilja, kaj je obvezno, kakšnega tipa so elementi in ali se lahko ponavljajo
- Za vsak podatkovni element obstaja niz, ki:
 - o Definira ime in referenčno oznako
 - o Oznako o tem, ali je obvezen, opciji ali pogojen
 - o Tip (numeričen, alfa-numeričen) in dolžina

RIP nudi nabor močno strukturiranih e-dokumentov in (togih) protokolov za poslovanje med organizacijami

Slabosti EDIFACT:

- Visoki stroški → počasna širitev
- Večinoma v B2B
- Toge osnove
- On-line ni mogoče prenašati pravil za procesiranje in ustreznih informacij
- Pokrivajo predvsem komunikacijo računalnik-računalnik, so strojno usmerjeni

Zahtevane spremembe:

- Večja fleksibilnost
- Rabi se podpora interakcije uporabnik - računalnik
- Procesiranje informacij

Evolucija RIP: danes kar precej uporabljan (v obliki XML)

(cca A št 183)

XML – eXtensible Markup Language

- Osredotoča se na podatke
- Omogoča definicije lastnih zaznamkov
- XML postaja osnova za RIP (EDI)

XML vs RIP

- XML: fleksibilni dokumenti in transakcije, zbirke podatkov, poizvedovanje, večpredstavnost, svetovni splet
- RIP: standardizirani poslovni dokumenti, poslovni običaji, celovita transakcijska podpora
- Sinergija XML in klasičnega RIP
- CML standardizira konzorcij W3C

XML

- Univerzalni jezik za podajanje podatkov (ločeni podatki in predstavitev in ločena informacija o strukturi podatkov)
- Pisan v človeku razumljivi obliki
- Jezik za podajanje podatkov, ne pa za procesiranje
- Tehnologija XML doda procesiranje

Dokument XML iz dveh osnovnih gradnikov

- Elementi
- Atributi

Elemente tvorijo začetne in končne značke in njihova vsebina

Npr <date>12/31/2001</dat>

Atributi pa tovrijo pare ime vrednost

<trial attempt=«1»>, kjer je attempt=«1» atribut, s tem da je attempt ime, »1« pa vrednost

Če ima podatek naslednika, mora biti element, ker atributi ne morajo imeti naslednikov

Osnovni pogoj za XML je njegova pravilna oblika. Dosežemo jo z označbami (markups) in podatkovnimi znaki (character data). Označbe se nanašajo na značke, podatkovni znaki ali vsebina pa so vsi ostali znaki.

Pravilno obliko preverjamo s sintaktičnim analizatorjem (parser).

XML dokument

Prva vrstica je deklaracija, kjer <?xml poda začetek deklaracija, version omogoča prepoznavanje različic, encoding omogoča uporabo različnih kodnih tabel, standalone določa, ali so dokumentu pridružene eksterne entitete.

DOCTYPE procesna instrukcija je opcionalna in z njo zahtevamo oveljavitev (validation) prek datoteke z vsebovanimi zahtevami o strukturi.

Vsek dokument vsebuje en sam korenski element. Vsi ostali morajo biti notri (vmeščeni med začetno in končno značko). Imena elementov se lahko začno s črko ali podčrtajem in so case sensitive in ne smejo vsebovati znakov za presledke.

Znaki space in tab se lahko pojavijo med podatkovnimi znaki, space, tab in linefeed med elementi pa so ignorirani.

Atributi imajo iste omejitve kot imena elementov

Tudi atributi morajo biti enolični.

V elementih (značkah) sta znaka LF in CR konvertirana v en sam presledek, v atributih pa je vsak tak znak pretvorjen v en presledek in tako lahko dobimo množico presledkov.

Nekateri definirani atributi

CDATA(character data) – teh podatkov razčlenjevalnik ne pregleduje, vključujejo pa alfa-numerične znake in presledke

ID, ki služi za oznako primarnega ključa (enoličen v dokumentu)

IDREF, ki se sklicuje na oznako primarnega ključa

Rezervirani znaki

< <- <

> >

& &

" »

' '

Za rezervirane znake lahko uporabimo CDATA, ker podatkov tega tipa razčlenjevalnik ne pregleduje

Atributi imenovani »procesna navodila«(processing instructions)

- Vsebujejo navodila, ki se predajo aplikaciji, ki procesira podatke
- Začno in končajo se z značkama <? In ?> ter vsebujejo cilj in navodilo samo
- Primer za procesiranje prikaza:

<?xmlstylesheet href="airlocks.xsl" type="text/xsl"?>.

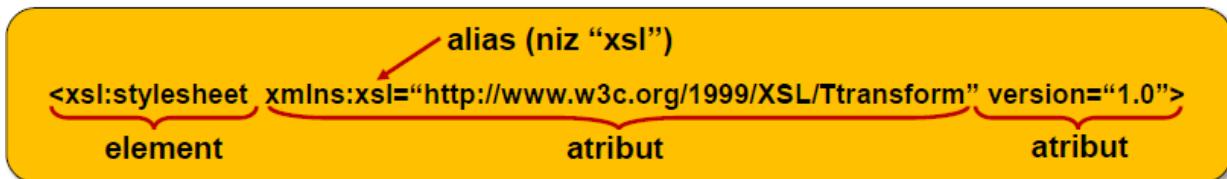
- Komentarji so med <! In -->

Pravilno oblikovan dokument ni nujno veljaven dokument

Pravilno oblikovan ustreza osnovnim zahtevam za XML, veljaven pa je skladen s podrobnimi definicijami, kateri elementi in atributi se lahko uporabijo, kje in kolikokrat. Veljavne dokumente dosežemo bodisi z definicijami tipov dokumenta DTD (data type definition) ali s shemami XML

Atributi »imenski prostor«

- Se spleča uporabit, zlasti ko obstaja možnost, da bo dokument združen z drugim dokumentom



DOM - Document Object Model

- Na XML gledamo kot drevo podatkov
- DOM podaja ustrezne definicije vmesnikov (apijev), za procesiranje teh podatkov

DOM ne definira množice objektov, ampak množico vmesnikov, da omogoči:

- Dostop do podatkov objektov
- Dostop do metod objektov
- Navedbo parametrov, ki jih lahko posredujemo predmetu in podatke s tipi, ki jih ta vrne

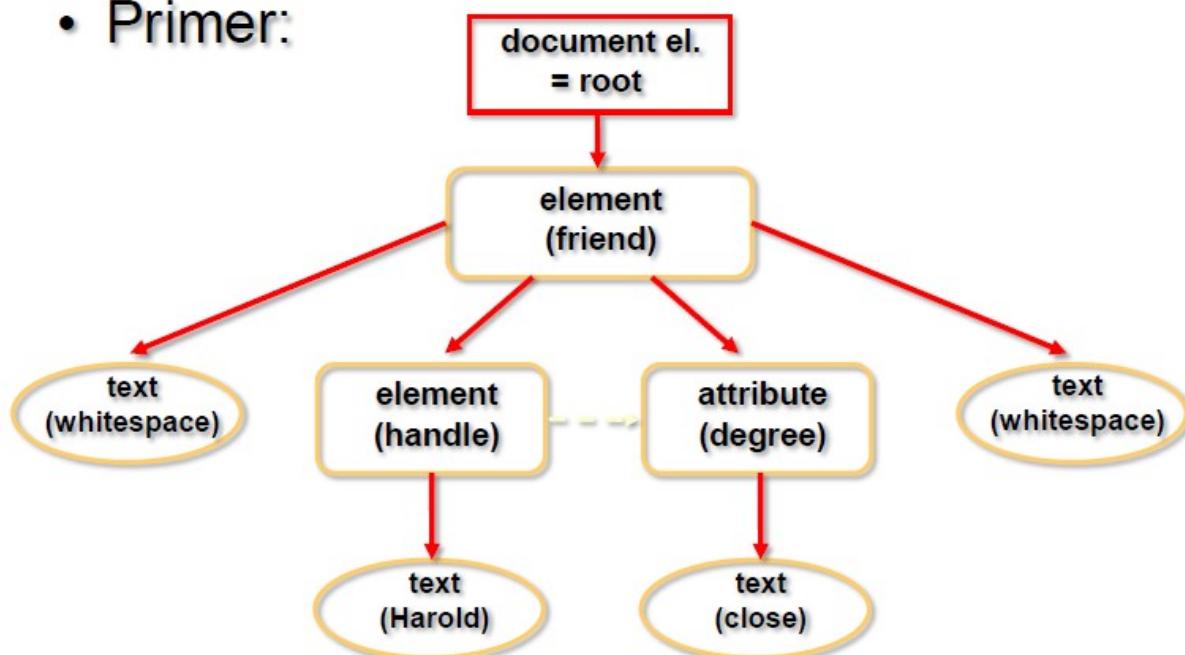
- Primer interpretacije po DOM:

```
<?xml version="1.0"?>
<friend>
    <handle degree="close">Harold</handle>
</friend>
```

- Imamo 8 vozlišč (nodes) različnih tipov:

- eno je vozlišče (celotnega) dokumenta,
- dve vozlišči sta z elementoma (vozlišče "friend" in njemu podrejeno vozlišče "handle"),
- eno vozlišče je z atributom (to je atribut "degree"),
- štiri vozlišča so s tekstrom ("close", "Harold" in znaka LF pred in za elementom "handle").

- Primer:



Nekateri vmesniki po specifikaciji DOM

- Document – predstavlja ves XML dokument, je koren
- Node – predstavlja vozlišče, vključuje metode kot je getNodeName, insertBefore in removeChild

- Element, ki je element dokumenta XML in je lahko povezan z vozlišči Attr ter vključuje metode kot je getElementsByTagName
- Attr vozlišče, ki ima lahko kot naslednika označki Text in EntityReference
- Text, ki je niz bre označb
- XDATASextion, ki zakrije označbe – označbe znotraj tega ne bodo dobile svojih vozlišč Element (prevajalnik jih ne vidi)
- DocumentType, vsebuje podatke, običajno podane v datotekah DTD ali pa podaja eksterno datoteko DTD
- Comment, Notation, EntityReference, ProcessingInstruction – vozlišča s podatki o njih samih

Vse v dokumentu je vozlišle – Node (Document, Element in njegov naslednik Text – vsi so Node-i)

Za dodajanje vsebine elementu je treba najprej narediti novo vozlišče z appendChild ali insertBefore

Z uporabo metod getChildNodes in getElementsByTagName lahko kreiramo seznam vozlišč in jih evalviramo

Dom in jezik XPath – nadgradnja XML

- XPath omogoča navigacijo po dokumentu
- Koren označuje z »/«, zatem pa imena elementov, ločena z »/«
- Primer »/friend/handle«
- DOM vs. XPath – Xpath le podaja pot, DOM pa lahko z njimi manipulira. DOM je vmesnik, XPath pa specifikacija jezika.

END OF SKLOP A

Validacija XML

- Lahko jo vršimo s pomočjo definicij tipov dokumenta (DTD) ali pa schem (XML Schema)
- Najbolj osnoven način za deklaracijo tipov dokumenta je uporaba definicije DOCTYPE (vrstica s procesno instrukcijo):
 - o Ta konstrukt uporabimo na začetku datoteke in določa dovoljeno strukturo le-te
 - o Ta konstrukt je lahko vceloti prisoten v izvorni datoteki XML ali pa (delno) v kaki drugi datoteki

- Primer - datoteka “votes.xml”:

```

<?xml version="1.0"?>
<!DOCTYPE votes [
    <!-- Definicije so navedene v tem delu. -->
]>
<votes totalVotes="10">
    <voter personid="Worker1" status="perm">
        <vote>against</vote>
        <comments>likesReps</comments>
    ...

```

- DOCTYPE konstrukt specificira elemente dokumenta (zgoraj je korenski element *votes* - njegova struktura še ni definirana).

- Primer DTD.

```

<?xml version="1.0"?>
<!DOCTYPE votes [
    <!ELEMENT votes (voter*)>
    <!ELEMENT voter (vote, comments)>
    <!ELEMENT vote (#PCDATA)>
    <!ELEMENT comments (#PCDATA)>
]

```

The diagram shows two yellow arrows originating from the text "parsed character data" and pointing to the "#PCDATA" declarations within the DTD. One arrow points to the declaration for the "vote" element, and the other points to the declaration for the "comments" element.

- Element *votes* lahko vsebuje nič ali več elementov *voter* (to podaja znak *), ta pa mora vsebovati točno en element *vote*, ki mu sledi točno en element *comments* in to vse v podanem vrstnem redu.

Itd...

Eksterno podani DTD

Definicije tipov lahko podamo v eksterni datoteki.

- Za podajanje zunanje datoteke uporabljam identifikatorja **SYSTEM** ali **PUBLIC**, ki se pojavita v glavi dokumenta:

```
<?xml version="1.0"?>
<!DOCTYPE votes PUBLIC "-//IT Resort //Voting System" "votes.dtd">
<votes totalVotes="10">
```

...

- Namesto URI bi lahko navedli lokalno datoteko: <!DOCTYPE dobave SYSTEM "dobave.dtd">

Preverjanje veljavnosti (validacija):

- Osnovo predstavljajo sintaktični analizatorji SAX (Simple API for XML). Dokument XML obravnavajo kot niz dogodkov (DOM ga tretira kot drevesno strukturo)
 - o To je prednost pri velikih datotekah, ker ne potrebujemo veliko pomnilnika
- Razčlenjevalniki DOM so pogosto zgrajeni na osnovi SAX-a – pri DOM so dosegljivi vsi podatki, rabimo več pomnilnika

Preverjanje veljavnosti z uporabo schem

- Končni cilji identični kot pri DTD, a želimo nativno rešitev XML
- Osnova schem je uporaba jezika XML tudi za definiranje struktur dokumentov
- Aplikacije, razvite za DTD, je potrebno prilagoditi pri preverjanju s shemami

- Preverjanje veljavnosti z uporabo shem.
 - Pri shemi deklaracijski vrstici sledi instrukcijska vrstica:

```
<xsd:schema xmlns:xsd="http://www.w3c.org/2000/10/XMLSchema
version="1.0">
```
 - Elementi so definirani s sledečo sintakso:

```
<element name="..." />
```
 - in imajo lahko več atributov kot je npr. tip
 type="..."
– ali omejitve glede kardinalnosti
 minOccurs="x", maxOccurs="x",
– kjer je „x“ naravno število ali nič.
 - Primer: <element name="koza" minOccurs="1"/>
- Preverjanje veljavnosti z uporabo shem.
 - Atributi so definirani s sledečo sintakso:

```
<attribute name="..." />
```
 - in imajo lahko več lastnosti kot je npr. tip
 type="..."
– ali privzete vrednosti
 use="x"
– kjer je x “optional”, “required” ali “default”.
 - Primer:

```
<attribute name="id" type="ID" use="required"/>
```

Primer sheme:

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3c.org/2001/XMLSchema">
  <xsd:element name="votes" type="voteType">
    <xsd:complexType name="voteType">
      <xsd:sequence>
        <xsd:element name="voter" type="voterType"
                     minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name="totalVotes" type="xsd:integer" />
    </xsd:complexType>
  <xsd:complexType name="voterType">
    <xsd:sequence>
      <xsd:element name="vote" type="xsd:string" />
      <xsd:element name="comments" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="personid" type="xsd:string" />
    <xsd:attribute name="status" type="xsd:string" />
  </xsd:complexType>
</xsd:schema>
```

Sheme XML

- Je XML dokument s korenskim elementom »schema«, kjer se nahajajo tudi ustrezne deklaracije imenskega prostora
- Omogočajo specifikacijo oz dokumentiranje in to ne samo podatkov v zvezi s samo shemo, ampak tudi podatkov, ki jih uporablja aplikacija, ki procesira shemo
- Predponi xs in xsd sta običajni

- Shema XML (predpona xs označuje podatke, ki tvorijo shemo).
 - Primer z dokumentacijo in podatki za aplikacijo.
- ```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3c.org/2001/XMLSchema">
 <xs:annotation>
 <xs:documentation>
 This document describes the schema for some inventory.
 In this phase no namespaces are used for the data.
 </xs:documentation>
 <xs:appinfo>
 <config xmlns="http://www.example.org/externalapp">
 <customertype>jazz culture fan</customertype>
 <destination id="gate A23" />
 </config>
 </xs:appinfo>
 </xs:annotation>
</xs:schema>
```

Uporaba sheme

Da shema postane efektivna, jo moramo povezati s konkretnim primerom dokumenta, ki mora biti skladen z njo. Da bi razčlenjevalnik (parser) našel shemo, pregleda instrukcijsko vrstico primerka dokumenta

```
<?xml version="1.0" encoding="UTF-8"?>
<collection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="collectibles.xsd">
</collection>
```

xsi: predpona je običajna  
(XML Schema instance)

Da bi parser prepoznal ukaze, morajo ti pripadati ustreznemu imenskemu prostoru. V zgornjem primeru atribut podaja kar lokalno datoteko.

Enostavni elementi in sheme

- Osnovni elementi XML so različnih tipov, ki jih je v shemi potrebno opredeliti

- Izhodišče so enostavni tipi (t.j. elementi brez atributov) in nekateri od njih so vnaprej definirani:

- tip *byte*, vrednost od -128 do 127;
- tip *unsignedByte*, 0 do 255;
- tip *integer*, od minus do plus neskončno;
- tip *int*, od -2.147.483.648 do 2.147.483.647;
- tip *unsignedInt*, od 0 do 4.294.967.295;
- ...

- **Enostavni elementi in sheme.**

```
<?xml version="1.0" encoding="UTF-8"?>
<xsschema xmlns:xs="http://www.w3c.org/2001/XMLSchema">
 <xs:annotation>
 ...
 </xs:annotation>
 <xs:element name="description" type="xs:string"/>
 <xs:element name="originDate" type="xs:date"/>
 <xs:element name="numOwners" type="xs:positiveInteger"/>
 </xsschema>
```

Element definiramo s ključno besedo element, kjer z uporabo atributov opredelimo ime novega elementa in tip podatkov, ki jih vsebuje

Vidna prednost pred DTD je npr. možnost omejitve dovoljenih vrednosti

Tako kot pri DTD lahko tudi pri shemah določimo privzete in fiksne vrednosti

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3c.org/2001/XMLSchema">
...
<xs:element name="numOwners" type="xs:integer"/>
<xs:element name="obtainable" type="xs:string" fixed="yes"/>
<xs:element name="originalOwner" type="xs:string" default="unknown"/>
</xs:schema>

```

- Zgornja definicija povzroči konverzijo elementa “obtainable” in elementa “originalOwner” v <obtainable>yes</obtainable> in <originalOwner>unknown</originalOwner> pred nadalnjim razčlenjevanjem.

Kompleksni elementi

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3c.org/2001/XMLSchema">
...
<xs:element name="obtainable" type="xs:string" fixed="yes" />
<xs:element name="originalOwner" type="xs:string" default="unknown" />
<xs:element name="collection">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="toys" type="xs:string"/>
 <xs:element name="furniture" type="xs:string"/>
 <xs:element name="pottery" type="xs:string"/>
 <xs:element name="autographs" type="xs:string"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
</xs:schema>

```

Zaradi elementa sequence se morajo elem. Pojaviti v istem zaporedju (če bi uporabili element all, bi bilo zaporedje poljubno, prav tako njihovo število, pri elem. Choice pa bi bilo obvezno izbrati en in samo en element).

```

<?xml version="1.0" encoding="UTF-8"?>
<collection xmlns:xsi="http://www.w3c.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="colectibles.xsd">
 ...
 <toys></toys>
 <furniture></furniture>
 <pottery></pottery>
 <autographs></autographs>
</collection>

```



osnovni dokument

Dodajanje atributov v shemo

- Podobno kot dodajanje elementov
- Če ima element attribute, je avtomatično kompleksnega tipa, enako kot če bi imel naslednike
- Atribute definiramo prek elementa attribute
- Primer dodajanja atributa elementu item

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3c.org/2001/XMLSchema">

<xs:element name="item">
 <xs:complexType> ← A complex type element is an XML element
 <xs:sequence> that contains other elements and/or attributes.
 <xs:group ref="itemCommon"/>
 </xs:sequence>
 <xs:attribute name="itemid" type="xs:integer"/>
 <xs:attribute name="keeper" type="xs:integer"/>
 <xs:attribute name="demand" type="xs:integer"/>
 <xs:attribute name="legal" type="xs:integer"/>
 </xs:complexType>
</xs:element> Mixed complexType element can contain
<xs:element name="collection"> attributes, elements and text.
 <xs:complexType mixed="false">
 <xs:choice minOccurs="1" maxOccurs="unbounded">
 <xs:element name="toys">
 <xs:sequence>
 <xs:element ref="item" minOccurs="0" maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>

```

```

<xs:element name="pottery">
 <xs:complexType mixed="true">
 <xs:sequence>
 <xs:element name="item" minOccurs="0" maxOccurs="unbounded">
 <xs:complexType>
 <xs:choice>
 <xs:group ref="itemCommon"/>
 <xs:element name="material" type="xs:string"/>
 </xs:choice>
 <xs:attribute name="itemid" type="xs:integer"/>
 <xs:attribute name="keeper" type="xs:integer"/>
 <xs:attribute name="demand" type="xs:integer"/>
 <xs:attribute name="legal" type="xs:integer"/>
 </xs:complexType>
 </xs:element>
 </xs:sequence>
 </xs:complexType>
</xs:element>

</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>

```

Atributi lahko uporabljajo vgrajene tipe

Definicija atributa elementa je na vrsti nazadnje, po definiciji naslednikov

Sheme vs. DTD – povzetek

- Sheme omogočajo specifikacijo dokumentov XML z večjo možnostjo nadzora in uporabo vgrajenih ter lastnih (od uporabnika definiranih) tipov
- Omogočajo tvorjenje novih tipov z razširitvami ali omejitvami obstoječih tipov
- Omogočajo točno podati, kolikokrat se element pojavi (ne pa samo ničkrat, enkrat ali večkrat)
- Možno je kombinirati podatke, ker prek uporabe imenski prostorov lahko ohranimo integriteto

Extensible Stylesheet Language (Transformations) – XSL(T)

- Podpora pretvorbi ene strukture v drugo
- Opravlja preslikave datotek XML v druge zapise: druge XML datoteke, ASCII (tekst), HTML, itd.

- Z uporabo predlog XSLT (XSLT templates) lahko pri pretvorbi tvorimo nove elemente in atributte iz obstoječih virov.

XSLT – predloga stile (stylesheet) vsebuje podatke za transformacijo, a je ne izvaja. Transformacijo opravlja aplikacija, ki jo krmili predloga stila

```
<?xml version="1.0"?>
<xsl:transform xmlns:xsl="http://www.w3c.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
<html>
 <head><title>Thank you ... </title></head>
 <body> <p>Dear customer,</p>
 <p>Thank you for your loyalty.</p>
 <p>Sincerely, IFF staff</p>
 </body>
</html>
</xsl:template>
</xsl:transform>
```

- **XSLT.**

- Osnovni dokument XML (test.xml):

```
<?xml version="1.0" encoding="UTF-8"?>
<customer>
 <customerName>Janez Kranjc</customerName>
</customer>
```

- Modificirana pola za transformacijo (test.xsl):

```
<?xml version="1.0"?>
<xsl:transform xmlns:xsl="http://www.w3c.org/1999/XSL/Transform" version="1.0">
<xsl:template match="customer">
<html>
 <head><title>Thank you ... </title></head>
 <body> <p> <xsl:value-of select="customerName"/>,</p>
 <p>Thank you for your loyalty.</p>
 <p>Sincerely, IFF staff</p>
 </body>
</html>
</xsl:template>
</xsl:transform>
```

Element *value-of* pove procesorju, da je treba vrednost prikazati, atribut *select* pa, katerega elementa vrednost je to.



## SEMANTIČNI SPLET (B, št 249)

Večina podatkov na internetu ni za naprave, glavnina je šibko strukturirana.

Danes se strojno iskanje znanja tipično začne s pregledovanjem podatkov, nato pridobivanjem informacij iz njih in nato sledi – znanje.

### Semantični splet (SmS)

Služi avtomatizaciji procesa:

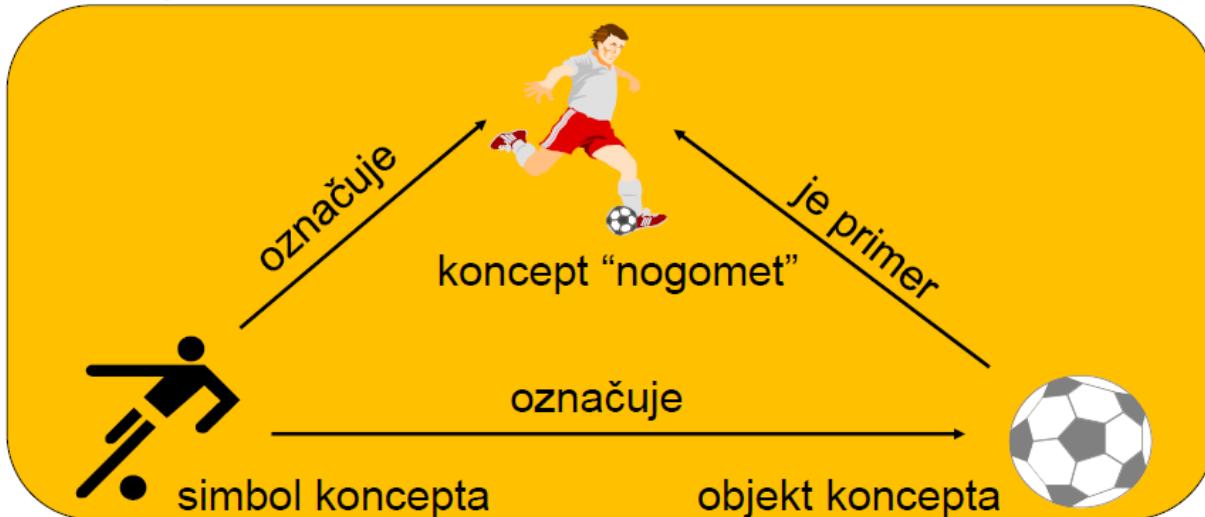
- Znanje se organizira v konceptualne prostore glede na njegovo naravo (pomen in namen)
- Orodja za avtomatizirano obdelavo bodo lahko samodejno pridobivala znanje, skrbela za odkrivanje nekonsistentnosti, itd.
- Iskanje po ključnih besedah bo (je) nadomeščeno z iskanjem prek povedi (ali besednih zvez).
- Tudi nadzor dostopa je lahko podprt

SmS primeren tudi za segment B2B

- Tipično pokrit z rešitvami RIP
- Kar je implementiranega z XML se lahko nadgradi po naravni poti, saj je SmS evolucija tehnologije XML
- S pomočjo SmS je moč definirati abstraktne domenske modele in realizirati ustreerne prevajalne sklope za avtomatizirano izmenjavo podatkov.
- V tem procesu bodo lahko sodelovali tudi umetni inteligentni agenti (pogajanja, koordinacija, itd.).

Osrednji element SmS so ontologije

- Koncept - kaj nekaj je, lastnosti tega „nekaj“. Objekti - primerki koncepta. Simbol - oznaka konceptov in objektov.



V računalništvu in informatiki definiramo ontologijo kot eksplizitno izraženo in formalno podano konceptualizacijo problemske domene.

Basically poveš kaj je kaj in v kakšnimi relacijami z drugimi stvarmi, da vsi dojemajo kot isto, da ni nesoglasij, nesporazumov (podjetja delajo slovarje, da vsi mislijo isto reč).

Ontologija sestoji iz končnega nabora izrazov in relacij med njimi. Izrazi podajo koncepte, relacije pa njihovo hierarhijo. Poleg tega imajo lahko ontologije še lastnosti reakcij, omejitve vrednosti in logične povezave med objekti.

Povzetek: SmS predstavlja interdisciplinarno področje na presečišču spletnih tehnologij (predvsem XML-a) in umetne inteligence.

#### Struktura Sms

- XML podaja sintaksos za strukturiranje, a brez semantičnih elementov. Shema XML je jezik za omejevanje sprejemljivih dokumentov.
- Okvir Resource Description Framework (RDF) je podatkovni model za sredstva RDF, to je objekte in relacije med njimi (modeli podani z XML). Shema RDF je jezik za podajanje lastnosti objektov in lastnosti ter semantiko za generalizacijo hierarhij objektov in relacij.
- Web Ontology Language (OWL) je napram RDF bogatejši za slovar za opis lastnosti razredov in relacij. Podamo lahko kardinalnost (npr. točno en, en ali več, ...), preseke (npr. 2 razreda nimata preseka), druge lastnosti (npr. simetričnost relacije).

Osnovni elementi SmS so torej:

- Jezik XML, shema XML
- RDF, shema RDF
- OWL

Logika je komplement za realizacijo SmS. S formalnim jezikom lahko izrazimo neko znanje.

- Podajmo enostaven primer pridobitve eksplizitnega znanja iz implicitno vsebovanega znanja – vzemimo, da vemo, da so vsi profesorji del pedagoškega osebja in to osebje del zaposlenih fakultete.
- Naj bo dr. Knez profesor:
  - $\text{prof}(X) \rightarrow \text{pedagogi}(X)$ ,  $\text{pedagogi}(X) \rightarrow \text{zaposleni}(X)$ ,  
 $\text{prof}(\text{knez})$
- Na podlagi zapisanega lahko sklepamo:
  - $\text{pedagogi}(\text{knez})$ ,  $\text{zaposleni}(\text{knez})$ ,  $\text{prof}(X) \rightarrow \text{zaposleni}(X)$
- Logika lahko služi tudi umetnim agentom, da na njeni osnovi sprejemajo odločitve, npr.:
  - $\text{lojalnaStranka}(X) \rightarrow \text{popust}(5\%)$

Poznamo več vrst logike, vendar bolj kot je izrazna, bolj računsko zahtevna je. Pri semantičnem spletu so podatki še v osnovi v precej omejeni obliki  $\rightarrow$  lahko uporabljamo enostavnejše vrste logike.

#### Arhitektura SmS

- Postopek temelji na ločevanju funkcionalnosti na samostojne sklope, »sloje«
- Umetni inteligentni agent, ki je sposoben uporabljati sloj neposredno pod samo lahko razume tudi druge nižje nivojske sloje in vsaj delno razume sloj oz. sloje nad sabo.

## RDF

- Je jezik metapodatkov
- Je neodvisen od domene
- Podaja osnove za kodiranje, izmenjavo in uporabo metapodatkov, ki omogočajo ekstrakcijo in obdelavo znanja (podatki -> informacije -> znanje)
- Strukturiran in strojno usmerjen jezik, tako da podpira interoperabilnost v smislu izmenjave podatkov na način oz. v obliki, ki omogoča računalniško obdelavo
- Čeprav se označuje kot le jezik je v bistvu podatkovni model
- Osnovni gradnik je trojica subjekt-predikat-objekt (tdui resource, property statement = property value)
- Interpretacija: <subject> has property <predicate> whose value is <object>
- Za konkretno predstavitev tega abstraktnega modela se uporablja sintakso XML

- **Resource Description Framework–RDF.**

- Vzemimo primer, kjer imamo resurs (**fotoaparat Nikon D70 SLR**), ki je enolično opredeljen s sledečim URI (oz. URL):

<http://www.yuchen.net/photography/SLR#Nikon-D70>

- Prvi del (obarvan oranžno) podaja imenski prostor (ki ni nujno, da dejansko obstaja).
- Razmejilni simbol (angl. delimiter) je znak #, ki loči imenski prostor od poimenovanja resursa znotraj tega imenskega prostora.  
Sintaksa je torej:

namespace || # || localResourceName

- Nadaljevanje primera – sedaj vpeljemo lastnost sredstva oz. resursa (vidik, relacijo, karakteristiko, ...), ki opredeljuje resurs:

<http://www.yuchen.net/photography/SLR#weight>

- Nadaljevanje primera – vpeljemo še trditev:

<http://www.yuchen.net/photography/SLR#Nikon-D70>  
has a

<http://www.yuchen.net/photography/SLR#weight>  
whose value is 1.4 kg

(<subject> has a property <predicate> whose value is <object>)

- Nadaljevanje primera – poenostavimo zapis z vzdevekom (aliasom) za imen. prostor:

xmlns:mySLR=<http://www.yuchen.net/photography/SLR#>

- Sedaj strukturo RDF lahko podamo kot  
mySLR:Nikon-D70 has a mySLR:weight whose  
value is 1.4 kg

- To predstavitev bi lahko še lepše podali v  
npr. tabelarični obliki.

- Izgradimo primer nabora trditev (navedb), ki jih podamo tabelarično:

Subject	Predicate	Object
mySLR:Nikon-D70	mySLR:weight	1.4 kg
mySLR:Nikon-D70	mySLR:pixel	6.1 M
mySLR:Nikon-D50	mySLR:weight	1.1 kg

- Če imamo vprašanje „Kakšne so lastnosti aparata Nikon D70?“ ga lahko v psevdokodi podamo kot

question.subject = mySLR:Nikon-D70

question.predicate = mySLR:\*

in dobimo odgovor: mySLR:weight, mySLR:pixel.

- Resource Description Framework – RDF.

- Primer dokumenta RDF:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns="http://www.yuchen.net/photography/Camera#">
 <SLR rdf:ID="Nikon-D70">
 <weight>1.4 kg</weight>
 </SLR>
</rdf:RDF>
```

- Prva (deklerativna) vrstica označuje dokument XML, druga pa podaja, da je to dokument RDF (ključna beseda RDF) in navaja pridružena imenska prostora, kjer je slednji imenski prostor privzet - če neko ime nima podanega im. prostora, se privzame ta imenski prostor (primer je SLR).

- Interpretacija dokumenta:
  - Dokument RDF podaja resurs,
  - katerega ime (identifikator) je Nikon-D70,
  - ta pa je primerek razreda SLR
  - in ima težo 1,4 kg.

Še ful tega RDF-ja...

Prednosti RDF

- RDF je fokusiran na standardizirane ontologije in s tem omogoča globalno enolično podajanje informacij na način, ki je strojno usmerjen.

+ kao druge

RDF shema – RDFS

- Sam RDF ne pove veliko o razredih, podrazredih, razmerjih med razredi, lastnostih, spremenljivkah, ki pripadajo razredom, itd.
- RDFS omogoča kreiranje ustreznih slovarjev, ki to zaobsežejo.
- RDFS doda semantiko resursom s tem, ko definira pomen nekega izraza prek objektov in lastnosti.
- RDFS je razširitev RDF in prav tako deluje nad trojicami oz. usmerjenimi grafi.

- RDF shema in ključni elementi.
  - Ključni elementi sheme so (osrednji) razredi, (osrednje) lastnosti in (osrednje) omejitve.
  - Osrednji razredi so rdfs:Resource, rdf:Property, rdfs:Class in rdf:Datatype.
  - Osrednje lastnosti so rdfs:subClassOf, rdfs:subPropertyOf.
  - Osrednje omejitve so rdfs:domain in rdfs:range.
  - V domeni RDFS ima izraz „razred“ isti pomen kot „resurs“ in se dostikratrabita ta izraza kot sinonima.

```
// Camera.rdfs
//
1: <?xml version="1.0"?>
2: <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3: xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4: xml:base="http://www.yuchen.net/photography/Camera.rdfs">
5: <rdf:Description rdf:ID="Camera">
6: <rdf:type
7: rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
8: </rdf:Description>
</rdf:RDF>
```

## i splet

- RDF shema – analiza enostavnega slovarja.
  - Prva vrstica je deklarativna vrstica datoteke XML.
  - Vsa vsebina dokumenta je med značkama <rdf:RDF> in </rdf:RDF> in ti sta skupni tako osnovnim dokumentom RDF kot tudi schemam.
  - Druge tri vrstice podajajo imenske prostore, od katerih tretja uporablja izraz „base“, zato se pripadajoč imenski prostor ne konča z „#“, ker je razširitveno pravilo URI v tem primeru oblike

xml:base || # || rdf:ID value

- Od pete do šeste vrstice je poimenovan razred Camera prek rdf:ID, da gre za razred pa podaja vrstica 6 in oznaka rdf:type.
- Te vrstice interpretiramo kot „razred Camera, ki je definiran v tej shemi, je podrazred rdfs:Resource“.
- Ker smo definirali razred brez uporabe lastnosti rdfs:subClassOf, je predpostavljeno, da je ta razred podrazred razreda rdfs:Resource, ki je korenski razred za vse razrede.
- Bolj intuitivno to podamo brez uporabe rdf:type, a uporabimo izraz rdfs:Class, da definiramo razred in značko rdf:ID, da ga poimenujemo Camera.

Še ful tega o RDFS.

OWL

Nadgradnja RDFS?

Doda še nov razred owl:Class za definicijo razredov v dokumentih OWL. Hierarhija vrhovnih razredov (po vrsti) je owl:Thing, rdfs:Resource, rdfs:Class, owl:Class

- Definirajmo vrhovni razred Camera za naš primer fotoaparatov:

```
<owl:Class rdf:ID="Camera">
 </owl:Class>
```

- Ekvivalentno lahko to podamo kot

```
<owl:Class rdf:ID="Camera">
 <rdfs:subClassOf
 rdfs:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>|
```

- Definicija razredov za domeno fotoaparatov:

```
// Camera.owl
//
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xml:base="http://www.yuchen.net/photography/Camera.owl">
 <owl:Class rdf:ID="Camera">
 </owl:Class>
 <owl:Class rdf:ID="Person">
 </owl:Class>
 <owl:Class rdf:ID="Digital">
 <rdfs:subClassOf rdf:resource="#Camera"/>
 </owl:Class>
 ...
 <owl:Class rdf:ID="Specifications">
 </owl:Class>
</rdf:RDF>
```

- Jezik OWL.

- Primer (owl:Restriction podaja anonimen razred):

```
<owl:Class rdf:ID="ExpensiveSLR">
 <rdfs:subClassOf rdf:resource="#SLR"/>
 <rdfs:subClassOf>
 <owl:Restriction>
 <owl:onProperty rdf:resource="#owned_by"/>
 <owl:allValuesFrom rdf:resource="#Professional"/>
 </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

- Interpretacija: Tu podajamo razred Expensive SLR, ki je podrazred SLR in ima lastnost owned\_by in kjer le primerek razreda Professional lahko nastopa kot vrednost.

- Predpostavimo, da agent, seznanjen s podano ontologijo, naleti na sledeč dokument:

```
<ExpensiveSLR rdf:ID="Nikon D200">
 <owned_by rdf:resource="http://www.yuchen.net/people#Liyang"/>
 <owned_by rdf:resource="http://www.yuchen.net/people#Jin"/>
</ExpensiveSLR>
```

- Agent bo lahko sklepal, da sta oba, Jin in Liyang Professionals (torej nista amaterja).
- Možno je podati izjeme, da imajo nekateri amaterji (poleg vsaj enega profesionalca) kljub vsemu tak aparat z owl:SomeValuesFrom:
 

```
<owl:someValuesFrom rdf:resource="#Professional"/>
```
- Dejstvo (znanje), da obstajajo drage kamere SLR, ki so v glavnem v lasti profesionalcev, lahko podamo tudi z definiranjem lastnosti expensiveOrNot, ki ima lahko vrednosti niza:

```
<owl:DatatypeProperty rdf:ID="expensiveOrNot">
 <rdfs:domain rdf:resource="#Digital"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

- Niza (strings) še nista definirana – naj bosta to niza expensive in inexpensive. Dobimo definicijsko datoteko, prikazano na naslednji prosojnici.

Itd...

Spletne storitve (SS)

Storitveno usmerjene arhitekture (Services Oriented Architectures, SOAs)

SOA poudarjajo poslovni vidik tehnologije

SOA in spletne storitve (Web services, WS)

- So nabor od platform neodvisnih tehnologij

- Omogočajo dostop do omrežnih storitev
- Povezujejo računalnike, zbirke podatkov, aplikacije in omrežja v virtualno strukturo, do katere je omogočen poenoten in standardiziran dostop
- Struktura je šibko sklopljena
- Osnovo predstavlja jezik (tehnologija) XML
- Želja je podpreti tudi trženje storitev

Konzorcija za standardizacijo je OASIS

#### Simple Object Access Protocol – SOAP

- Je protokol, ki omogoča programu z enega op. Sistema komunikacijo s programom na drugem op. Sistemu z uporabo XML preko http
- SOAP določa način kodiranja glave http in datoteke XML za zagotovitev interoperabilnosti

#### Web Services Description Language (WSDL)

- Temelji na jeziku XML
- Podaja storitve, ki so na voljo v omrežju
- Podaja način dostopa (vmesnike)

Universal Description, Discovery and Integration (UDDI) za enotno registracijo in iskanje storitev na spletu  
→ zadeva je propadla

Spletne storitve omogočajo aplikaciji poklicati drugo aplikacijo in na predvidljiv način poslati ter prejeti podatke

Spletne storitve delimo na: tiste, ki vrnejo odziv in tiste, ki ga ne, a povzročijo, da se izvede neko opravilo

Format za izmenjavo sporočil običajno SOAP, lahko pa je tudi drug, npr. MIME

Protokol za storitve je običajno http, lahko pa je tudi drug, npr. e-pošta SMTP

#### Spletne storitve so aplikacije

- Kí temeljijo na XML
- Kjer si strežniki in odjemalci izmenjujejo ustrezeno formatirana sporočila

- Kjer aplikacije uporabljajo ustreerne protokole za medsebojno komunikacijo
- In kjer obstaja infrastruktura za iskanje storitev in poizvedbo o njihovih detajlih

Ključne besede: SOAP, WSDL, UDDI

- **Primer sporočila SOAP:**
  - storitev, ki posreduje nivo kisika na neki lokaciji;
  - storitev interpretira sporočilo kot klic metode `getOxygenPercentage(3, "B")`.

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3c.org/1999/XMLSchema">
<soap-env:Header/>
<soap-env:Body>
 <ns1:getOxygenPercentage xmlns:ns1="LifeSupportInformationSystem"
 soap-env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
 <deckId xmlns:xsi="http://www.w3c.org/1999/XMLSchema-instance"
 xsi:type="xsd:int">3</deckId>
 <section xmlns:xsi="http://www.w3c.org/1999/XMLSchema-instance"
 xsi:type="xsd:string">B</section>
 </ns1:getOxygenPercentage>
</soap-env:Body>
</soap-env:Envelope>
```

- **Primer sporočila SOAP.**
  - Rezultat prejšnje metode je vrednost.
  - Vrednost je vsebovana v spodnjem sporočilu SOAP.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsi="http://www.w3c.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3c.org/1999/XMLSchema">
<SOAP-ENV:Body>
 <ns1:getOxygenPercentageResponse xmlns:ns1="LifeSupportInformationSystem"
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
 <return xsi:type="xsd:double">0.62</return>
 </ns1:getOxygenPercentageResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Kako najti želeno spletno storitev

- Sistematično iskanje po IP-jih in vratih je vse prej kot optimalno

- Uvedba registrov, kjer organizacije vnesejo podatke o sebi (poslovnih procesih) in storitvah, ki jih nudijo
- Registri dostopni prek UDDI, vendar standard ni zaživel
- Podatki o storitvah so podani v jeziku WSDL (Web Services Description Language) v ustreznih datotekah.

#### Realizacija spletne storitve

- Potrebujemo ustrezen spletni strežnik
- Potrebujemo orodja za procesiranje sporočil SOAP

- **Primer enostavne spletne storitve.**
  - Storitev za orbitalno postajo, ki preverja vrednosti veličin, pomembnih za življenje:
    - imamo na voljo metode (za odčitavanje veličin),
    - vsaka metoda potrebuje vsaj en parameter in vrne enostavno vrednost.
  - Vsakič, ko prebere vrednosti, aplikacija pokliče tudi metodo, ki zabeleži čas nazadnje odčitanih vrednosti.
  - Opis navedene storitve podamo z ustreznou datoteko WSDL.

- Enostavna spletna storitev - 1. del.

```
import java.util.GregorianCalendar;
import java.util.Date;

public class LifeSupportInfo {
 GregorianCalendar lastCheckedOxygen = new GregorianCalendar();
 GregorianCalendar lastCheckedTemperature = new GregorianCalendar();
 public double getOxygenPercentage (int deckId, String section) {
 if (deckId==3){
 if (section.equals("B")){
 lastCheckedOxygen.setTime(new Date());
 return .62;
 } else if (section.equals("C")) {
 lastCheckedOxygen.setTime(new Date());
 return .68;
 } else {
 return 0;
 }
 } else {
 return 0;
 }
 }
}
```

- Enostavna spletna storitev - 2.del.

```
public int getAvgHullTemperature (String side) {
 lastCheckedTemperature.setTime(new Date());
 return 45;
}
public GregorianCalendar getLastChecked(String systemType) {
 if (systemType.equals("oxygen")) {
 return lastCheckedOxygen;
 } else if (systemType.equals("temperature")) {
 return lastCheckedTemperature;
 } else {
 return null;
 }
}
```

– Program, ki bo predstavljal našo storitev, prevedemo, da dobimo datoteko **.class**, ki jo potem interpretiramo z JVM.

Še nekaj o proxijih je vmes.

- Primer enostavne storitve.
  - Opisni dokument (angl. deployment descriptor file) – to je (psevdo) WSDL:

```
<isd:service xmlns:isd="http://xml.apache.org/xml-soap/deployment"
 id="urn:LifeSupportInformationSystem">
 <isd:provider type="java" scope="Application"
 methods="getOxygenPercentage getAvgHullTemperature getLastChecked">
 <isd:java class="LifeSupportInfo" static="false"/>
 </isd:provider>
 <isd:faultListener>org.apache.soap.server.DOMFaultListener</isd:faultListener>
</isd:service>
```
  - Označene so metode, ki so na voljo, "id" je identifikator storitve, vršilec storitve pa je razred LifeSupportInfo.class.

Uporaba protokola SOAP za klic storitve

- Sporočilo SOAP tvori ovojnica, ki je sestavljena iz glave in telesa (header, body), vsi ti elementi pripadajo imenskemu prostoru

<http://schemas.xmlsoap.org/soap/envelope>.

- Element telo predstavlja podatke (koristno vsebino, payload)
- Če pošiljamo sporočilo SOAP prek http uporabljamo metodo POST
- Koraki:
  - o Kreiramo sporočilo
  - o Vzpostavimo povezavo
  - o Pošljemo sporočilo
  - o Prejmemo odgovor
  - o Preberemo odgovor

Web Services Description Language – WSDL

- Ta jezik omogoča, da lahko neko storitev uporablja tudi drugi uporabniki omrežja

- Z njim podamo lokacijo storitve, razpoložljive operacije, format sporočil, parametre in njih tipe ter opredelimo vrnjene podatke
- To vse podamo v dokumentu, ki je v bistvu sestavljen iz hierarhije deskriptorjev

## ➤ Najprej definiramo niz, kjer

- > podamo način povezave (npr. HTTP ali SMTP),
- > podamo vrata, ki opredeljujejo končne točke ali URL,
- > podamo metode in z njimi vhodna ter izhodna sporočila, vključujoč sporočila o napakah.

## ➤ Tipi sporočil diktirajo podatke, ki gredo od / k strežniku in tipično uporabljajo sheme XML.

UDDI

- Glavni objekt (= predmet) v registru je *businessEntity*, ki opiše poslovno entiteto:
  - > podaja njeni ime,
  - > podaja njen opis,
  - > navede kontaktne točke,
  - > vsebuje več tipov dodatnih identifikatorjev, npr. davčno številko, identifikator DUNS (za preverjanje plačilne sposobnosti) itd.
- Vse te podatke pridružimo enemu ključu - univerzalno enoličnemu identifikatorju (Universally Unique Identifier - UUID).

- Podatki o storitvah so shranjeni v objektu tModel, npr. kazalec na datoteko WSDL.
- UDDI vmesnik (API) nudi:
  - metode *find*, ki omogočajo iskanje objekta brez poznavanja ključa: *find\_business*, *find\_relatedBusiness*, *find\_binding*, *find\_service*, *find\_tModel*;
  - metode *get*, ki omogočajo s pomočjo ključa pridobiti podake o objektu: *get\_bindingDetail*, *get\_businessDetail*, *get\_businessDetailExt*, *get\_serviceDetail*, *get\_tModelDetail*.
- Vse metode se realizirajo s pomočjo protokola (sporočil) SOAP.

#### Business Process Execution Language – BPEL

- Omogoča kreiranje kompozitnih storitev z dodatno logiko za koordiniranje osnovnih storitev, pri čemer se naslanja na vmesnike WSDL

Poenostavljena obravnava BPEL, ki temelji na standardih 1.1 in 2.0

- Primer uporabe – poslovni scenarij:
  - Želimo realizirati paket storitev za nekega kupca, ki bi vključeval dobavo, namestitev (inštalacijo) ter vzdrževanje strojnih in programskih komponent IS skupaj z njihovo nadgradnjo.
  - Preden tak paket storitev ponudimo stranki mora biti ponudba validirana (ali so določeni elementi storitve na voljo na izbrani lokaciji, ali so kompatibilni s kupčevim informacijskim sistemom in našim „know-how“-om, itd).
- V ta namen kreiramo proces BPEL.

- **Realizacija poslovnega scenarija z BPEL:**
  - Kreiramo SS, ki sprejme naročilo kupca.
  - Ta SS razdeli celotno naročilo na podsklope in jih pošlje v obdelavo specializiranim SS (npr. SS, ki skrbi za upravljanje strojne opreme in SS, ki skrbi za upravljanje programske opreme).
  - Po obdelavi specializirani SS vrneta odgovor, nakar glavna SS vrne odziv kupcu.
  - Zatem proces BPEL (točneje - glavna SS) ostane v stanju pričakovanja strankinega naročila.
- **BPEL je v bistvu tehnologija, ki omogoča procesno integracijo IS prek SS.**
- **Podpira dva temeljna prisotpa k povezovanju storitev in sicer**
  - orkestracijo, kjer osrednji proces (lahko tudi SS) koordinira v kompozitno storitev vključene SS (te se „ne zavedajo“, da sodelujejo v višje nivojskem storitvenem kompozitu);
  - koreografijo, kjer ni centralnega koordinatorja, zato se morajo vključene SS „zavedati“ procesa, v katerem sodelujejo (to je operacij, ki jih morajo izvršiti, sporočil, ki si jih morajo izmenjati in časovnih zaporedij).

Poslovni procesi so v BPEL opisani kot izvršljivi procesi ali pa kot abstraktni poslovni protokoli

BPEL specificira, kako vključimo storitve (zaporedno ali vzporedno), pogoje, zanke, deklaracije sprem. In pritejanje njihovih vrednosti, ...

Osnova procesa BPEL so aktivnosti, ki se delijo na elementarne (primitivne) in strukturirane.

Elementarne aktivnosti uporabljajo konstrukte:

- Za vključitev druge SS – konstrukt <invoke>
- Za čakanje odziva druge SS - <receive>
- Za generiranje odziva za sinhrono operacijo - <reply>
- Za pripomaganje vrednosti sprem. - <assign>
- Za oznanjanje napak - <throw>
- Za čakanje - <wait>
- Za zaključek celotnega procesa - <terminate>

Kombiniranje – elementarne konstrukte kombiniramo na algoritičen način s strukturiranimi aktivnostmi

- Zaporedne aktivnosti - <sequence>
- Vzporedne aktivnosti - <flow>
- Vejitve - <switch>
- Zanke - <while>
- Izbiro alternative - <pick>

Vsek BPEL definira še spremenljivke (<variable>) in pa partnerske povezave (<partnerLink>).

Počasi več slajdov na to temo.

END OF SKLOP B

Spletne storitve – visokonivojski pregled

- Poleg BPEL imamo še jezik BPMN (Business Process Model and Notation), ki je novejšega datuma
- Je grafični jezik za opisovanje poslovnih procesov
- Jezika sta komplementarna – včasih je proces laže podati preko BPEL, včasih pa zgolj grafično

- Spletne storitve – visokonivojski pregled.

podatki / protokol / jezik	standardizirano s strani	namen
XML	W3C	predstavitev podatkov
SOAP	OASIS	klic omrežnih storitev
REST	arhitekturni slog	klic omrežnih storitev
WSDL	W3C	opis vmesnikov in funkcij
UDDI	OASIS	avt. reg. in odkr. storitev
ebXML	OASIS, UN/CEFACT	avtom. reg. in odkrivanje procesov
BPEL	OASIS	kompoziti storitev

Inteligentni agenti (IA) so uveljavljena tehnologija, ki izvira iz umetne inteligence.

Inteligencia – nivo sposobnosti agenta, da se zaveda svojega stanja in okolice

- Odzivnost – reakcije na okolico
- Prilagodljivost
- Iniciativnost – ustvarjalno zaznavanje sprememb in sledenje prilagojenemu cilju

Posredništvo – stopnja avtonomije, tj. nivo potrebe po uporabnikovi intervenciji

Mobilnost – sposobnost izvedbe nalog v različnih okoljih

AI pri e-poslovanju

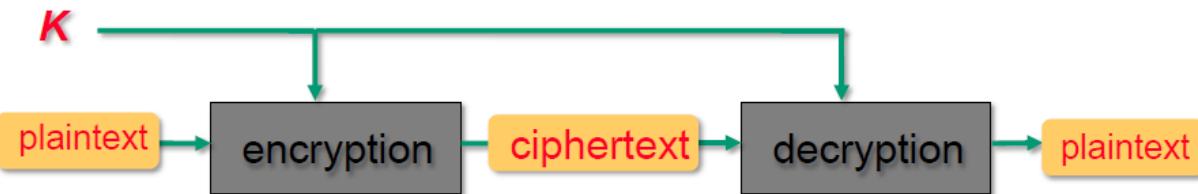
- Izvajanje poslovnih transakcij
- Boljša prilagodljivost storitev strankam
- Pregledovanje tržišča
- Učinkovito zbiranje informacij v imenu strank
- Izvajanje poslovnih simulacij
- ...

- Uporaba pri poslovanju:
  - V določeni domeni (poslovnom okolju) opravimo analizo poslovnega procesa, kjer opredelimo probleme, identificiramo željene rešitve ter vloge udeležencev.
  - V drugem koraku identificiramo potrebne agente, storitve in ontologije, kar predstavlja dizajniranje konkretno agentne strukture.
  - Sledi realizacija procesa v okviru agentne platforme, kjer kreiramo agente, jih ustrezno organiziramo in koordiniramo.

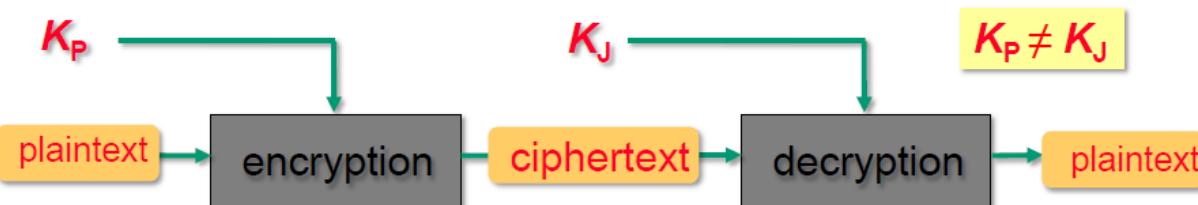
#### Varnost e-poslovanja

- Varnostni mehanizmi – kriptografski algoritmi, mehanizmi fizičnega varovanja
- Varnostne storitve: kriptografski protokoli
- Podsistemi: overitelji (certification authorities, CAs), imenik, časovna normala, agencija za upravljanje imenskega prostora
- Organizacijski visiki: varnostna politika
- Pravni vidiki: zakonodaja

- Simetrični kriptosistem – en ključ.



- Asimetrični kriptosistem – dva ključa.



Asimetrična kriptografija -> certifikati -> overitelji

Digitalno podpisani e-dokument in običajen papirnat dokument sta pravno-formalno izenačena (or something)

Ena od osnov za digitalno podpisovanje so enosmerne zgoščevalne funkcije (EZF).

- Naj bi bilo nemogoče (računsko neobvladljivo) najti  $x$ , če veš  $y$  in  $f(x) = y$
- Nemogoče za  $x_1$  najti  $x_2$  da  $f(x_1) = f(x_2)$

Izračun verjetnosti in vrste kolizij

### – Vrste kolizij:

- “Fixed-date paradox” → zamenjava vnaprej definiranega teksta z modificiranim.
- “Birthday paradox” → zamenjava poljubnega neškodljivega teksta s škodljivim.

- Izračun verjetnosti.
  - Imejmo množici  $A$  in  $B$ , kjer ima  $A$  neskončno,  $B$  pa končno moč.
  - Elementi množice  $B$  naj bodo naravna števila z zaprtrega intervala  $[1, n]$ .
  - Verjetnost preslikave poljubnega elementa množice  $A$  v določen element v  $B$  je enaka za vse elemente  $B$  (uniformna porazdelitev).
  - Koliko elementov (“ $m$ ”) iz množice  $A$  moramo naključno izbrati, da pride do vsaj ene kolizije v  $B$  pri določeni verjetnosti  $p$ ?
- Izračun verjetnosti:
  - “ $c$ ” pomeni kolizijo,
  - “ $cf$ ” pomeni ne-kolizijo (ni trka),
  - “ $n$ ” je število elementov v množici  $B$ .
- Paradoks fiksnega dne (“Fixed day paradox”):
 
$$p_c(n, m) = 1 - p_{cf}(n, m) = 1 - (n-1)^{m-1} / n^{m-1}$$
- Paradoks rojstnega dne (“Birthday paradox”):
 
$$\begin{aligned} p_c(n, m) &= 1 - p_{cf}(n, m) = \\ &= 1 - (n/n)_1 * ((n-1)/n)_2 * \dots * ((n-m+1)/n)_m = \\ &= 1 - n! / (n^m (n-m)!) \end{aligned}$$

- Primer napada na digitalni podpis (osnova je paradoks rojstnega dne).
  - Dear [prof/dr] Johnson, this is to express my [honest/sincere] opinion about [teaching qualifications / academic qualifications] of dr. Kent. His [excellent / outstanding] achievements in his field with [numerous / many] references ...
  - Dear [prof/dr] Johnson, this is to express my [honest/sincere] opinion about [teaching qualifications / academic qualifications] of dr. Kent. His [poor / weak] achievements in his field with [irrelevant/ minor] references ...

Preprečevanje (zmanjševanje) kolizij pa to

- Preprečevanje redundantne topologije:
  - minimizacija števila variabilnih polj.
- Preprečevanje redundančnih vrednosti posameznega polja:
  - minimalen nabor - inkrementalne vrednosti.
- Preprečevanje topološkega prekrivanja:
  - [The signer] [orders] [the transfer] [of US\$] [X1] [from account #] [Y1] [to account #] [Z1.]
  - [The signer] [confirms] [the receipt] [of US\$] [X2] [from account #] [Y2] [to account #] [Z2.]

- Uporaba semantične ekvivalence ob različni sintaksi – namerna omejena redundanca na nivoju sporočil:
  - [The signer] John M. [agrees with the withdrawal of] [\$100] [as a monthly fee for CT Bank services.]
  - [A monthly fee for CT Bank services in amount of] [\$100] [is confirmed by signer] [John M.] [ to be withdrawn from his account.]
- Trenutno primerne EZF:
  - SHA-256/384/512 (SHA-3 – skrivna vrata).
  - RIPEMD -160, dolžina 160 bitov, avtorji Dobbertin, Bosselaers, Preneel.
- Neprimerne EZF:
  - SHA-1 (od leta 2010 dalje).
  - MD 2 / 4 / 5, avtor R. Rivest (prvi napadi na MD5 so bili delo prof. dr. H. Dobbertina iz l. 1996).

Bitcoin

Pametne kartice

Trendi (RFID/NFC)

Tudn na 529 je zamaknjen e v naslovnu ...

## Poslovne funkcije in arhitektura IS e-poslovanja

- Osnova je integracija aplikacij na nivoju organizacije
  - o Obvladovanje :
    - o prvin poslovnih procesov
    - o odnosov s strankami
    - o kadrov
    - o nabavne funkcije (verige dobav)
    - o prodajne funkcije (prodajne verige)

Glavna ponudnika programske opreme:

Nemški SAP in ameriški ORACLE

- Ključen element pri poslovanju so pravočasne in na pravi način dostopne informacije ob upoštevanju:
  - konsistentnosti podatkov;
  - porazdeljevanju in omogočanju enostavnega dostopa do informacij vsem zaposlenim (“sploščevanje” hierarhij).



Obvladovanje prvin poslovnih procesov (OPPP)

- Administrativno usmerjeni IS
- Realiziramo z rešitvami Enterprise Resource Planning, ERP

OPPP je vezivo, ki omogoča izmenjavo podatkov med oddelki podjetja s poenotenjem ustreznih procesov

Uvajanje ERP (SOPPP)

- Občutljivo področje
  - o Nujno sodelovanje vodstva
  - o Projektov ne sme voditi samo osebje iz oddelkov IT
  - o Izhajati moramo iz dobrega razumevanja poslovnih procesov, nato poiskati načine optimizacije in končno stvari realizirati z IT
- Napačen pristop lahko privede do resnih težav

Obvaldovanje odnosov s strankami (OOS) (Customer Relationship Management – CRM)

Lažje prodajaš starim strankam, slabo izkušnjo stranke posredujejo naprej.

3 face OOS:



- Proces prodaje – podpora različnih pristopov:
  - Prodaja komplementarnih ali nadgrajenih proizvodov in storitev (angl. cross-sell, up-sell).
  - Prodaja s segmentiranjem in s pomočjo prožilcev (izpolnjeni določeni pogoji).
- Neposredno trženje.
  - Avtomatizacija in integracija zaradi možnosti sledenja stroškov preko množice kanalov in prilagajanja uporabniku.
  - Učinkovito posredovanje informacij stranki (hiter odziv, ustrezna predstavitev).
- Proces podpore (posredna in neposredna).
  - Učinkovito upravljanje zahtev za storitev, dogоворов o realizacijah, pogodbah, stikih.
  - Uspešno reševanje težav:
    - identifikacija problemov,
    - izbira prave kontaktne točke,
    - sledenje odpravljanju težave,
    - neposredna podpora pri stranki.
- Proces navezave:
  - diferenciacija strank glede na obstoječe podatke (to je glede na pretekle transakcije).

Načrt uvedbe – glavna vodila

- Definiraj vizijo
- Razumi stranko

- Analiziraj svojo združbo
  - Oceni pripravljenost združbe in položaj konkurence
  - Vzpostavi merljive cilje
  - Ovrednoti aplikacijo s poudarkom na integraciji
- **Načrt uvedbe – podpora vodila.**
    - Poglej na stvari s perspektive stranke.
    - Identificiraj in agresivno osvajaj čimprej dosegljive podcilje.
    - Za krovno upravljanje projekta delegiraj enega samega ravnatelja (managerja).
    - Implementiraj postopoma in upoštevaj spremembe.
    - Kreiraj okolje, ki je napram stranki enotno ne glede na točko interakcije s podjetjem.

#### Obvladovanje verige prodaje (OVP)

- Integrirana strategija pridobivanja naročil
- Za kupca enostaven proces
- Povečevanje učinkovitosti prodaje
- Koordiniranje dela prodajne ekipe

OVP = “The right product / service to the right customer for the right price through the right channel at the right time”.

#### Ključni dejavniki

- Močan prodor samopostrežnega pristopa
- Stroški predprodajne podpore
- Povečevanje kompleksnosti proizvodov in storitev

- naraščajoča možnost večjega števila napak pri naročilih;
- spreminjanje prodajnih kanalov;
- rast podjetij, ki je posledica združevanja (fuzije) ali prevzemov.

- Oblikovanje cen je kompleksen proces

#### Obvladovanje verige dobav – OVD (Supply Chain Management)

- OVD je koordinacija informacijskih, finančnih in materialnih tokov med sodelujočimi partnerji pri izdelavi določenega proizvoda ali vršitvi storitve.



- Faze pri uvajanju OVD:
  - Definicija ciljev.
  - Ocena organizacije z vidika zrelosti za uvedbo OVD.
  - Izdelava taktičnega načrta in opredelitev načinov (mer) vrednotenja.
  - Ustanovitev projektne skupine.
  - Integracija partnerjev.
  - Merjenje (vrednotenje) uspešnosti.
  - Izobraževanje.
  - Načrt obvladovanja neuspešnih potez.

## Obvladovanje nabavne funkcije – ONF

- Združbe porabijo 50%-60% zaslužka za plačilo proizvodov in storitev, potrebnih za lastno nemoteno delo.
- Nabava vs. preskrba: nabava pomeni le neposredne aktivnosti, povezane z nakupom proizvoda ali storitve (povpraševanje in plačilo), preskrba pa poleg nabave vključuje tudi transport, skladiščenje in vročevanje.
- Uporabljamo enoten izraz - nabava.
- Dve vrsti nabave.
  - Nabava, povezana neposredno z osnovno dejavnostjo – elementi in lastnosti:
    - surovine, komponente;
    - čas preskrbe določa proizvodni proces;
    - ni (kompleksnih) procesov odobritve;
    - visoka avtomatizacija.
  - Nabava, posredno povezana z osnovno dejavnostjo – elementi in lastnosti:
    - vzdrževalni in nadomestni deli, potni stroški, pisarniški material, računalniška oprema;
    - čas ni strogo definiran;
    - zahteva odobritev;
    - slaba možnost avtomatizacije.
- Ključni dejavniki za uspešno ONF.
  - Možnost nadzora s strani managementa:
    - kateri proizvodi oz. storitve so na voljo,
    - kje se dobijo omenjeni proizvodi oz. storitve,
    - kdo odobri naročilo.
  - Možnost izbora “on-line”:
    - zmanjšanje napak,
    - redukcija s papirjem povezanih operacij,
    - hitrejša izvedba.
  - Elektronsko naročanje:
    - ni pretipkavanja, avtomatska verifikacija,
    - ni papirja in klasičnega pošiljanja.

## Harmonizirane aplikacije za zajemanje znanja – HAZZ (Knowledge Tone Applications)

- Osredotočene na individualizirano podporo pri odločanju
- Stremimo k prijaznim aplikacijam
- **Ključna beseda – prilagodljivost:**
  - prilagodljivi filtri za podatke,
  - prilagodljivi prožilci,
  - fleksibilni in intuitivni vmesniki za izdelavo poročil,
  - kar se podpira uporaba naravnega jezika.
- **Klasična paradigma sistemov za podporo odločanju je „vprašanje – odziv“.**
- **Paradigma pri HAZZ pa je često obrnjena – “odziv – vprašanje“.** To je inverzna logika, saj iščemo ozadje odziva, kaj je torej vzrok.
- **HAZZ ni proizvod kot tak, ampak poslovni okvir oziroma koncept, ki sodi na področje sistemov za podporo odločanja.**

## Formalne metode (FM)

So metode, ki temeljijo na logiki in diskretni matematiki

Področja namembnosti: načrtovanje, podajanje in izvedba računalniških sistemov

Zahtevajo jasno navedbo predpostavk in sklepanja na podlagi relativno majhne množice pravil

Prednost uporabe FM je zmanjševanje takih vrst sklepanja, ki so posledica intuicije. Nadalje, omogočajo računsko preverjanje specifikacij kode – primerno za načrtovanje in analizo rešitev IT.

- **Formalna specifikacija** pomeni opis sistema v formalnem jeziku.
  - Formalni jezik je določen z elementarnimi simboli (primitivi) in produkcijskim pravili.
  - Opis sistema predstavlja definicije, nabor aksiomov in rezultati izpeljav skupaj s pravili sklepanja.
  - Rezultate izpeljav predstavljajo vse formule, izhajajoče iz definicij in aksiomov ob upoštevanju pravil sklepanja (formalna dedukcija).
  - Postopki izpeljav skupaj z dokazi (množico izpeljanih formul) pa predstavljajo teorijo.

## Dvojen vidik FM

- Teorija dokaza – podaja semantični vidik, je neinterpretiran račun
- Teorija modela – interpretacijo imenujemo model, če so vsi aksiomi v tej interpretaciji resnični in pravila transformacije pri dokazovanju ohranjajo resničnost

## Formalna analiza:

- Jo pogosto izvajamo s pomočjo programskega orodja, ki omogoča:
  - o Preverjanje in
  - o Razbroščevanje podane formalne specifikacije
- Ustrezna orodja uporabljajo pristope, ki tipično slonijo na:
  - o Neposredni izvedbi (simulaciji, animaciji)
  - o Izpeljavi teoremov in preverjanju dokazov

## Nekateri predstavniki formalnih metod

- Jezik Z (»zed«): FM, ki temelji na teoriji množic in predikatnem računu prvega reda
- Logika BAN: FM osnovana na modalni logiki
- Jezik UML: Ni prava FM, temelji na objektno usmerjenih modelirnih tehnikah
- Možnost strogog formalnega dokazovanja: Z (teorija dokaza in modela), BAN (teorija dokaza), UML (je ni)

## Formalen jezik Z

### **Oslove jezika Z**

**Z** je formalen jezik, namenjen specifikaciji in dizajniranju (predvsem softverskih) sistemov. Razvit je bil na Univerzi v Oxfordu in je v standardiziran s strani organizacije ISO:

- Omogoča specifikacijo sistemov na abstraktni predstavitevno-proceduralni ravni. Za predstavitevno abstrakcijo, ki vključuje matematične prvine (množice, relacije, ...), ni pomembno *kako*, temveč *kaj* želimo realizirati.
- Tako gramatika kot semantika izhajata iz klasičnih matematičnih osnov: teorije množic in predikatnega računa, kjer predikati lahko vsebujejo eksistencialne in univerzalne kvantifikatorje.
- Od matematičnih osnov se razlikuje po strogem določeni sintaksi (formalni matematični pristopi tovrstni standardizaciji niso bili nikoli podvrženi).

## Elementi specifikacije v jeziku Z

- Specifikacija v **Z**-ju je sestavljena iz enot, imenovanih *scheme*, ki so interno razdeljene na dve polovici:
  - v zgornjem delu se nahajajo deklaracije,
  - v spodnjem delu so predikati, ki podrobno opredeljujejo zgoraj deklarirane elemente (ta del je lahko tudi prazen, brez omejitev).

*Shema*  
—  
*deklaracije*  
—  
*predikati*

- Deklaracije definirajo imena spremenljivk in morajo vsebovati tudi tipe le-teh.
- Predikati lahko vsebujejo univerzalne in eksistencialne kvantifikatorje.
- Osnovne tipe podajamo v oglatih oklepajih.

[TIP]

- Aksiomatske definicije podajamo na sledeč način:

$\frac{Maksimum : \mathbb{N}}{Maksimum \leq 255}$

- Generične definicije podajamo s shemami, ki imajo dvojno zgorno črto.

$\frac{[A, B]}{\frac{First : A \times B \rightarrow A}{\forall a : A; b : B \bullet First(a, b) = a}}$

## Operacije in spremembe stanj

- Relacije podajajo povezave med vhodom in izhodom ter opredeljujojo začetna in končna stanja spremenljivk pri operacijah.
- Primer operacije potenciranja: shema *Potenciranje*, kjer je vhod celo pozitivno število ali nič, označeno s spremenljivko *in?*, in izhod celo število, označeno s spremenljivko *out!*.

*Potenciranje*  
—  
*in?, out! : Z*  
—  
*in? ≥ 0*  
*out! = 3<sup>in?</sup>*

- Obrazložitev: prvi predikat zahteva, da je vhod število, večje ali enako nič, drugi pa, da je izhod potenza 3 in vhoda.
- Vhodne spremenljivke se končujejo z vprašajem, izhodne pa s klicajem.

## Modularnost jezika

- Vključevanje shem - s pomočjo sheme *PrvaSchema* vpeljimo novo shemo in obe skupaj sta ekvivalentni tretji shemi:

<i>PrvaSchema</i>	_____
$x, y : \mathbb{N}$	
$U : \mathbb{P}\mathbb{N}$	
$x \leq y$	
<i>DrugSchema</i>	_____
<i>PrvaSchema</i>	
$V : \mathbb{P}\mathbb{N}$	
$x \in V$	
<i>TretjaSchema</i>	_____
$x, y : \mathbb{N}$	
$U, V : \mathbb{P}\mathbb{N}$	
$x \leq y$	
$x \in V$	

## Modeliranje dinamičnega vidika sistemov

- Dinamične relacije med elementi pri spremembah stanja sistema ponazorimo tako, da imenu elementa, ki se po operaciji spremeni, dodamo opuščaj.
- Notacija  $\Delta Stanje$  pomeni vse spremenljivke iz sheme *Stanje* skupaj z njihovimi novimi vrednostmi (spremenljivke z opuščaji).

$\Delta Schema$	_____
<i>Schema</i>	
<i>Schema'</i>	

- Če operacije ne spremenijo stanja (spremenljivk), potem namesto  $\Delta$  pišemo  $\Xi$  (glej primer na naslednji strani).

## Tipi, deklaracije in kvantifikatorji

- Poleg standardnih privzetih tipov (naravna števila, cela števila...) lahko uvedemo lastne generične tipe:  
 $[EvropskaUnija]$
- S pomočjo tipov izvajamo deklaracije, kajti vsaka spremenljivka je oz. mora pripadati določenemu tipu:  
 $nemcija, francija : EvropskaUnija \quad x, y : \mathbb{N}$
- V predikatih uporabljamo omejevalne in neomejevalne kvantifikatorje (tako za univerzalne, kot tudi eksistencialne kvantifikatorje):  
 $\forall x : \mathbb{N} \bullet x + 1 > x \quad \forall x : \mathbb{N} \mid x > 100 \bullet x^2 > 10000$
- Povezava med omejevalnimi in neomejevalnimi kvantifikatorji ( $D$  je deklaracija,  $P, Q$  pa formuli):  
 $(\forall D \mid P \bullet Q) \Leftrightarrow (\forall D \mid P \Rightarrow Q)$   
 $(\exists D \mid P \bullet Q) \Leftrightarrow (\exists D \mid P \wedge Q)$

## Množice, relacije in funkcije

- Množice definiramo z naborom elementov ali pa s pomočjo že obstoječih množic:

$$pijace == \{cockta, radenska\} \quad kvadrazi == \{n : \mathbb{N} \mid n * n\}$$

- Kartezijski produkt množic  $X$  in  $Y$  je množica vseh urejenih parov, kjer je prvi element para iz prve in drugi iz druge množice.

$$\{1, 3\} \times \{5\} = \{(1, 5), (3, 5)\}$$

- Relacije pojmujemo kot množice urejenih parov in torej predstavljajo podmnožice kartezijskega produkta:

$$X \leftrightarrow Y == \mathbb{P}(X \times Y)$$

- Funkcije so definirane kot posebna vrsta relacij, ki preslikajo element iz prve množice v na jveč en element druge množice. Primer:

$$X \rightarrow Y == \{f : X \rightarrow Y \mid (\forall x : X; y, z : Y \mid x \mapsto y \in f \wedge x \mapsto z \in f \Rightarrow y = z)\}$$

## Zaporedja in torbe

- Zaporedja so posebna vrsta funkcij, ki so definirana kot

$$\text{seq } X == \{f : \mathbb{N} \rightarrow X \mid \text{dom } f = 1.. \#f\}$$

kjer je  $X \rightarrow Y == \{f : X \rightarrow Y \mid \text{dom } f \in \mathbb{F} X\}$  in  $\mathbb{F} X$  predstavlja množico vseh končnih podmnožic  $X$ . Primeri zaporedij:

$$\langle jan, feb, dec, apr, maj \rangle \in \text{seq ImenaMesecov}$$

$$\langle \langle \rangle, \langle apr, maj \rangle \rangle \in \text{seq}(\text{seq ImenaMesecov})$$

- Nad zaporedji so definirane opereacije stika, filtriranja, ekstrakcije, itd. Primer operacije stika:

$$\langle jan, feb, dec, apr \rangle \cap \langle okt \rangle = \langle jan, feb, dec, apr, okt \rangle$$

- Torbe so strukture, podobne množicam, saj je vrstni red elementov nepomemben, je pa za razliko od množic pomembno število ponovitev elementa. Primer ekvivalentnega zapisa:

$$[a, b, a, c, c] \quad [b, a, c, a, c].$$

## Preimenovanje in skrivanje

Preimenovanje sheme pomeni preimenovanje prostih spremenljivk.

$$\frac{\begin{array}{c} S \\ \hline x, y : \mathbb{N} \\ V : \mathbb{P} \mathbb{N} \end{array}}{x \in V \Rightarrow y \in V}$$

Naj bo  $T \hat{=} S[m/x]$ , potem je rezultat naslednja shema:

$$\frac{\begin{array}{c} T \\ \hline m, y : \mathbb{N} \\ V : \mathbb{P} \mathbb{N} \end{array}}{m \in V \Rightarrow y \in V}$$

$T \hat{=} S \setminus (x)$  pomeni skrivanje spremenljivke  $x$  v shemi  $S$ :

$$\frac{\begin{array}{c} T \\ \hline y : \mathbb{N} \\ V : \mathbb{P} \mathbb{N} \end{array}}{\exists x : \mathbb{N} \bullet (x \in V \Rightarrow y \in V)}$$

## Kompozicija shem

Kompozicija je način povezovanje spremenljivk po spremembi stanja ene sheme s spremenljivkami pred spremembo stanja druge sheme. Te spremenljivke morajo imeti isto osnovno ime (to je ime, ki ostane po odstranitvi dekoracije).

Imejmo shemi  $S$  in  $T$  in tvorimo  $S \circ T$ :

1. Tvorimo  $S[s^+/s']$ .
2. Tvorimo  $T[s^+/s]$ .
3. Tvorimo konjunkcijo zgornjih dveh shem -  $S[s^+/s'] \wedge T[s^+/s]$ .
4. Tvorimo  $(S[s^+/s'] \wedge T[s^+/s]) \setminus (s^+) -$  skrijemo torej  $s^+$ .

## Kompozicija - primer

Imejmo shemi  $S$  in  $T$ :

$S$
$x?, s, s', y! : \mathbb{N}$
$s' = s - x?$
$y! = s$

$T$
$x?, s, s' : \mathbb{N}$
$s < x?$
$s' = s$

Po preimenovanjih (koraka 1 in 2) napravimo konjunkcijo:

$Konjunkcija Preimenovanih Shem$
$x?, s, s^+, s', y! : \mathbb{N}$
$s^+ = s - x?$
$y! = s$
$s^+ < x?$
$s' = s^+$

## Kompozicija - primer

V zadnjem koraku skrijemo spremenljivko  $s^+$ :

$Kompozicija Shem$
$x?, s, s', y! : \mathbb{N}$
$\exists s^+ : \mathbb{N} \bullet (s^+ = s - x? \wedge$
$y! = s \wedge$
$s^+ < x? \wedge$
$s' = s^+)$

Kompozicije uspemo velikokrat poenostaviti z uporabo zakona sintaktične ekvivalenze  $P[t/y] \dashv\vdash \exists y : Y \bullet P \wedge y = t$ , kjer je  $t$  istega tipa kot  $y$  in kjer je  $y$  prost v formuli  $P$ . Zato namesto

$$\exists s^+ : \mathbb{N} \bullet \underbrace{s^+ = s - x? \wedge y! = s \wedge s^+ < x? \wedge s^+ = s'}_{P}$$

v predikatnem delu lahko pišemo  $P$

$$s' = s - x? \wedge y! = s \wedge s' < x?.$$

## Common Criteria – CC

- Nabor standardov, osredotočen na evalvacijo zagotavljanja ustreznih nivojev varnosti strojne in programske opreme IS
- Ciljne skupine: proizvajalci (identifikacija var. Zahtev), potrošniki (specifikacija merljivih lastnosti in primerjav izdelkov) in evaluatorji (nadzor)

## Stopnje EAL (Evaluation Assurance Level)

- EAL1 obsega funkcionalno testirane proizvode z najnižjim nivojem zagotovil, kjer detektiramo očitne napake in zagotavlja varnost za okolja z elementarnimi varnostnimi zahtevami.
- EAL 2 obsega strukturalno testirane proizvode, kar še vedno lahko dosežemo brez dodatnih zahtev za razvijalce, če le-ta uporablja standarde v pričakovanem obsegu. EAL 2 zagotavlja nizko do zmerno stopnjo varnosti brez dodatnih zahtev za celotno sledenje razvoja glede varnosti.
- EAL3 obsega metodično testirane proizvode z uporabo varnostnih metod v fazi načrtovanja proizvodov brez potrebe po bistvenih spremembah razvojnih pristopov, to je reinženiringa in dodatnih stroškov. Zagotavlja zmerno stopnjo varnosti.
- EAL 4 – metodično načrtovani in testirani proizvodi, maksimalna varnost, ki je še dosegljiva z uporabo ustaljenih kakovostnih komercialnih pristopov. Ni potrebno bistveno vključevanje specialistov, je ekonomsko opravičljivo znotraj obstoječe proizvodnje, nudi zmerno do visoko stopnjo varnosti.
- EAL 5 (EAL 6) – pol-formalno (delno formalno) načrtovanje (verificiranje načrtovanja) in testiranje, znatna podpora s strani specialistov, visoka stopnja varnosti, opazen vpliv stroškov iz naslova varnosti.
- EAL 7 – formalno načrtovan in testiran proizvod, intenzivno sodelovanje ekspertov, proizvodi za ekstremno rizične namestitive oz. ščitenje najbolj občutljivih informacij, kar opravičuje visoke stroške za tovrstno opremo.

## Intelektualna lastnina

- E-poslovanje izpostavlja kreativnost in ustvarjalnost ter pomen znanja
- Ščitenje intelektualne lastnine (njen rastoč pomen, njene vrste, načini ščitenja)
- Razvoj podjetij v smeri "proizvodnje" intelekt. lastnine kot take – primer IBM-a:
  - spočetka izdeluje pisalne stroje,
  - nato računalniško strojno in programsko opremo,
  - nato večinoma programsko opremo in končno ter
  - se v času e-poslovanja začne preusmerjati v intelektualno lastnino ter postane "tovarna znanja" (od rač. do nano-tehnologij).
  - Leta 2004 je IBM prijavil 3248 patentov.
- Danes to počne tudi Google, Apple, itd.

Intelektualna lastnina pokriva ustvarjalne aktivnosti na industrijskem, znanstvenem in umetniškem področju.

Intelektualna lastnina je povezana z moralnimi (pravice avtorja nad storitvijo) in ekonomskimi vidiki (gospodarsko izkoriščanje in splošni družbeni napredek).

Ustvarjalno delo se realizira šele, ko je dostopno javnosti in s tem postane gibalo napredka

- Zakoni s področja zaščite intelektualne lastnine so temeljno orodje za varovanje te lastnine.
- Omenjeni zakoni določajo vsebinske, časovne in prostorske okvire varovanja:
  - Zakon o industrijski lastnini,
  - Zakon o avtorskih in sorodnih pravicah,
  - Zakon o varstvu topografije polp. vezij.
- Delitev (vrste) intelektualne lastnine.



- Industrijska lastnina.
  - Ustrezni zakon ščiti patent (pa patent s skrajšanim trajanjem), model in vzorec, blagovno in storitveno znamko ter označbo porekla blaga.
  - Upravičenci pridobijo tako materialne kot tudi moralne pravice:
    - Materialne pravice obsegajo izključno pravico gospodarskega izkoriščanja.
    - Moralna pravica ustvarjalcev je, da so navedeni v prijavi in vseh listinah (kot kreatorji).
- Vrste patentov.
  - Redni patent ali patent z normalnim trajanjem 20 let (to je osnovna oblika).
  - Patent s skrajšanim trajanjem – ta traja deset let, nosilec pa ni obvezan predložiti potrdila o novosti oz. popolnem preizkušu (ta patent ne more varovati postopka, rastlinske vrste ali živalske pasme).
  - Dopolnilni patent – če izumitelj po vložitvi temeljne prijave najde nove rešitve, lahko v roku 18 mesecev zahteva tudi dopolnilni patent.

UIL = urad za intelektualno lastnino

- Patent.
  - Patent predstavlja pogodbo med prijaviteljem in pristojno državno organizacijo (pri nas UIL RS).
  - Patent varuje izum, ki je nov, je dosežen z ustvarjalnim delom na ravni izumiteljstva in je uporabljiv.
  - Velja, da je izum nov, če ni bil predhodno dostopen javnosti z objavo (tu šteje tudi internet), razstavitevjo in če se razlikuje od vsebine prijav vloženih v RS ali tujini.

- Patent.
  - S patentom se ne da ščititi (znanstvenih) odkritij, teorij, matematičnih metod, računalniških programov in drugih pravil, načrtov, metod ter postopkov za duhovne aktivnosti - te ne štejejo med izume. Ampak...
  - Patent ne ščiti kirurškega ali diagnostičnega postopka ali postopka zdravljenja, ki se uporablja neposredno na živem človeškem ali živalskem telesu, razen izumov, ki se nanašajo na snov pri takem postopku.
- Patent.
  - Patent ne varuje izuma, ki je v nasprotju z zakonom ali moralu.
- Model in vzorec.
  - Z modelom se zavaruje vidna (zunanja) oblika industrijskega ali obrtnega izdelka.
  - Vzorec varuje vsakomur vidno novo sliko ali risbo, ki se da prenesti na industrijski ali obrtni izdelek.
  - Ne model ne vzorec ne more varovati fotografije, tehničnega načrta, kartografskega dela, skice.
- Blagovna in storitvena znamka.
  - S to znamko se zavaruje znak, ki je v gospodarskem prometu namenjen razlikovanju blaga oz. storitev.
  - Z znamko varovani znak ne izključuje pravice drugega, da uporablja enak ali podoben znak za označevanje druge vrste blaga ali storitev, razen če gre za sloveč znak ali znamko (!).
  - Z blagovno znamko se zavaruje slika, risba, beseda, izraz, vinjeta, šifra, kombinacija znakov ali barv.

- Kot znamka se ne more uporabljati znak, ki nasprotuje morali, ni primeren za razlikovanje blaga oz. storitev, ki označuje le vrsto blaga, njihov namen, čas ali način proizvodnje, kakovost, ceno, količino, težo, kraj oz. geografsko poreklo.
- Nadalje se ne more varovati znakov za označevanje posamezne vrste blaga ali storitev, znakov, ki lahko povzročijo zmedo v gospodarskem prometu ali pri potrošniku, znakov istovetnih že varovanim znakom, uradnih znakov ali državnih simbolov.
- Označba porekla blaga.
  - Z njo se zavaruje geografsko ime proizvodov, katerih posebne lastnosti so pretežno odvisne od kraja oziroma področja, na katerem so nastali in so te lastnosti naravnega porekla.
  - Področje uporabe obsega naravne proizvode, kmetijske pridelke, industrijske proizvode, obrtne izdelke.
  - Ta označba ne ščiti geografskega imena proizvodov, ki je postalo po dolgi uporabi splošno znano.

Pol pride postopek prijave

Rabiš zahtevo, opis, zahtevki, sliko/skico(če potrebno) in povzetek. Rabiš tudi pravnika.

+potrdilo o plačilu pristojbine.+ pooblastilo, če vloženo prek zastopnika.

- Patent – postopek prijave (nadaljevanje).
  - Vsak del patentne prijave se mora začeti na novem listu A4, uporablja se lahko le ena stran lista.
  - Opis, zahtevke in skice mora prijavitelj lastnoročno podpisati.
- Patent – postopek podelitev.
  - Urad na vsaki prijavi zabeleži datum in številko prejema prijave.
  - Najkasneje v 3 mesecih preveri, če je prijava v skladu z zakonom in eventualno zahteva dopolnitev.

Itd...

- **Prednostna pravica.**
  - Od datuma prejema pravilne prijave patenta, modela, vzorca ali znamke ima prijavitelj prednostno pravico nasproti vsakomur, ki za enak izum, obliko telesa, sliko ali risbo vloži prijavo kasneje.
  - Prednostna pravica za patent traja 12 mesecev, za model, vzorec in znamko pa 6 mesecev. V tem času ima prijavitelj prednostno prijavo pred konkurenco na področju vseh držav članic mednarodne unije za varstvo industrijske lastnine.
- **Sejemska pravica.**
  - Oseba, ki na uradno priznani mednarodni razstavi ali sejmu v RS ali drugi državi članici mednarodne Unije za varstvo industrijske lastnine razstavi izum, obliko telesa, risbo ali znak, lahko zahteva v treh mesecih prednostno pravico, pri čemer mora priložiti ustrezeno potrdilo.
  - Čas trajanja te pravice je 3 mesece.
  - Tovrstnih sejmov (razstav), ki omogočajo uveljavljanje te pravice, je na svetu malo.
- **Patentno varstvo v tujini.**
  - Osnova je nacionalna prijava v posamezni državi. Ta se najkasneje v letu dni (za trajanja prednostne pravice) po prvi prijavi vloži pri ustreznem organu želene države.
  - Postopek je podoben slovenskemu, a nekateri uradi izvedejo celoten postopek popolnega preizkusa pred samo podelitvijo patenta.
  - Tako ščitimo lastnino le v posamezni državi.

- Institut mednarodne prijave patenta – PCT:

- To poenostavi postopek patentne prijave v več državah (postopek se vodi v Ženevi).
- Prijavitelj lahko opravi eno mednarodno poizvedbo glede novosti (predhodnega preizkusa) za vse države, ki so članice Patent Cooperation Treaty – PCT (cca. 100 držav).
- Pozitiven rezultat poveča možnost dodelitve nacionalnih patentov, za čas preizkusa pa se odloži tudi pačilo stroškov prijave.

- Evropski patent velja v EU in državah, ki imajo z EPO sklenjene sporazume o razširitvi veljavnosti patentov:

- EPO opravi poizvedbo o stanju tehnike.
- V primeru ugovorov posameznih držav se lahko odstopi od nadaljnjega postopka v dani državi.
- Ko je evropski patent podeljen, je potreba v treh mesecih uradu ciljne države predložiti prevode podeljenega patentna in plačati pristojbino.

- **Avtorske in sorodne pravice.**

- Umetniško in inovativno - raziskovalno ter razvojno delo je vezano na predstave, objave rezultatov v člankih, kongresih, itd. – te dejavnosti so ščitene prek avtorskih in sorodnih pravic.
- Zakon o avtorskih in sorodnih pravicah pokriva
  - pravice avtorjev na njihovih delih s področja književnosti, znanosti in umetnosti;
  - pravice izvajalcev, proizvajalcev fonogramov, filmskih producentov, radijskih in TV organizacij ter založnikov (sorodne pravice);
  - individualno in kolektivno uveljavljanje avtorske in sorodnih pravic.

Pod avtorske pravice spadajo naši programi

- Avtorsko delo – pet ključnih pogojev:
  - Individualnost kot najpomembnejša lastnost (npr. dva slikarja slikata lahko isti motiv).
  - Intelektualnost, kar pomeni, da se v delu odraža človeški duh, misli, čustva.
  - Je lahko le rezultat človekovega ravnanja in zahteva določen ustvarjalni napor.
  - Ima izraženost v zunanjem svetu, tako da je zaznavno za človeške čute.
  - Pri tem je področje umetnosti in znanosti potrebno pojmovati zelo široko.
- Dela, ki niso avtorska:
  - ideje, načela in odkritja;
  - uradna besedila z zakonodajnega, upravnega in sodnega področja,
  - ljudske umetniške stvaritve.
- Avtorske pravice:
  - Avtorska pravica je enovita pravica na avtorskem delu, iz katere izvirajo moralne avtorske pravice, premoženske pravice in druge pravice.
  - Te pravice izhajajo iz same stvaritve dela (s tem, ko nastane npr. risba, je ustvarjeno tudi avtorsko delo).
  - Avtorska pravica je avtomatična in ne zahteva dodatnih formalnosti, lahko pa avtor zaradi lažjega dokazovanja dela arhivira oziroma vpiše v register (npr. pri Avtorski agenciji za Slovenijo).
  - Avtor je lahko le fizična oseba, pravica nastane z nastankom dela in traja še 70 let po smrti avtorja.
- Moralne avtorske pravice.
  - Varujejo avtorjeve duhovne in osebne vezi do dela.
  - Avtor ima izključno pravico odločiti o prvi objavi dela.
  - Pripada mu pravica do priznanja avtorstva, navedbe le-tega in oznake.
  - Brez privoljenja avtorja ni dopustno poseganje v delo.
  - Avtor ima pravico skesanja (možnost preklica materialne avtorske pravice ob resnih razlogih in povrnitvi škode).

Pol so stvari v zvezi s kršenjem itd.

Kultura združbe in varovanje podatkov

- Podatki so premoženje združbe, zato jih je potrebno zaščititi zaradi:
  - ohranjanja tržne prednosti;
  - zakonodajnih zahtev;
  - ohranjanja dobrega imena.
- Ščitenje informacij zagotovimo z nadzorom prek
  - postopkov,
  - organizacijske strukture in
  - softversko / hardverskih rešitev.

Varnost je proces, ne stanje

Varnost s pomočjo tehnologije mora biti komplementarno podprtzo ustreznim upravljanjem

Najceneje varovanje vključiti že med načrtovanjem.

Ključni standardi: ISO 27000, COBIT / ISACA in dopolnila

Avtentikacija in certificiranje – Qweb

- Certifikat se preverja dva do štirikrat obnavlja letno z oddaljene lokacije (eksterno preverjanje spletnega mesta), enkrat letno pa neposredno na lokaciji vršilca storitve (pregled samih poslovnih procesov, povezanih s spletnim poslovanjem).
- Pri nas tovrstno akreditacijo opravlja Slovenski inštitut za kakovost in meroslovje in imetnike akreditacije zabeleži v spletno dostopnem registru.

- Relevantna domača zakonodaja:
  - Zakon o e-poslovanju in e-podpisu.
  - Zakon o spremembah in dopolnitvah zakona o elektronskem poslovanju in elektronskem podpisu.
  - Uredba o pogojih za elektronsko poslovanje in elektronsko podpisovanje.
  - Zakon o avtorskih in sorodnih pravicah.
  - Zakon o industrijski lastnini.
  - Zakon o elektronskih komunikacijah.
  - Zakon o varstvu osebnih podatkov.
  - Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih.
  - Kazenski zakonik.
  - Zakon o prekrških.

#### Zakonodaja

- Začetna ideja - zakonodaja naj bi bila sprejeta na mednarodnem nivoju (npr. OZN).
- Zakon o elektronskem poslovanju in elektronskem podpisu (ZEPEP).
  - ZEPEP ureja elektronsko poslovanje, ki zajema poslovanje v elektronski obliki na daljavo z uporabo inf. in komunikacijske tehnologije in uporabo elektronskega podpisa v pravnem prometu, kar vključuje tudi elektronsko poslovanje v sodnih, upravnih in drugih podobnih postopkih.

## ZEPEP – načela

- Nediskriminacija elektronske oblike:
  - e-oblika ekvivalentna papirnati oblik;
  - e-podpisu ne smemo odreči veljavnosti, tudi če ne temelji na kvalificiranem potrdilu ali če ni oblikovan s sredstvom za varno e-podpisovanje.
- Odprtost in tehnološka nevtralnost:
  - neodvisnost od konkretno tehnologije (npr. kriptografskih principov).
- Načelo pogodbene svobode strank:
  - določbe, ki urejajo e-poslovanje, niso prisilne narave in stranke lahko uporabijo drugačne rešitve od predvidenih v zakonu;
  - stranke lahko razmerje pogodbeno uredijo tako, da izključijo zakonsko ureditev (besedna zveza "če ni dogovorjeno drugače"), razen ko gre za varovanje javnega interesa ali posebno varstvo ene pogodbene stranke (besedna zveza "stranka mora", "prepovedano je", itd.).
  - Načelo dvojnega pristopa
  - Načelo varstva osebnih podatkov
  - Načelo varstva potrošnikov
  - Načelo mednarodnega priznavanja

### ZEPEP – elektronska oblika

- Izenačenost s pisno, ko je el. Oblika doesgljiva tudi po daljšem času, so podatki primerni za kasnejšo uporabo + pri shranjevanju sporočil je potrebno hraniti vse v njem vsebovanje ali z njim povezane podatke (tudi izvor, čas pošiljanja in prejema...).
- Vsi e-podatki, ki so pravilno shranjeni in ustrezajo gornjim pogojem, so izvirniki
- Izključitev dokumentov, ki morajo biti sklenjeni v obliki notarskega zapisa in oporočni posli

Elektronsko sporočanje – sporočilo poslano ali prejeto na elektronski način

- Sporočila ni moč pripisati pošiljatelju:
  - če je imetnika sporočila pravočasno obvestil, da sporočilo ni njegovo;
  - če lahko z običajno skrbnostjo dobrega (ravnanje dobrega gospodarja) ugotovimo, da sporočilo ni pošiljateljevo;
  - če prejemnik z uporabo dogovorjene tehnologije in postopkov lahko ugotovi, da elektronsko sporočilo ni pošiljateljevo.
- Posrednik elektronskega sporočila je oseba, ki za drugo osebo pošle, hrani ali nudi storitve v zvezi z e-sporočilom; zaradi posredništva se ne šteje za pošiljatelja ali prejemnika.
- Podvajanje elektronskih sporočil:
  - prejemnik je upravičen šteti vsako prejeto elektronsko sporočilo kot posamično in ravnavati v skladu s tem;
  - izjema je, če je bilo sporočilo podvojeno in je prejemnik to vedel ali bi moral vedeti, če bi ravnal kot skrben gospodar ali če bi uporabil dogovorjeno tehnologijo in postopek.
- Potrditev prejema sporočila.
  - Veljavnost poslanega sporočila je odvisna od potrdila o prejemu, če
    - pošiljatelj pred pošiljanjem ali v samem sporočilu zahteva potrditev in
    - v sporočilu navede, da je e-sporočilo pogojeno s potrdilom o prejemu.
  - Prejemnik in pošiljatelj se lahko dogovorita tudi za posebno obliko potrjevanja sporočil, sicer kot potrdilo velja kakršnakoli samodejna potrditev ali ravnanje prejemnika, ki zadostuje, da pošiljatelj izve, da je bilo e-sporočilo prejeto.
  - Potrdilo o prejemu samo po sebi ne zagotavlja celovitosti prejetega sporočila.

Sporočilo poslano, ko vstopi v IS izven nadzora pošiljatelja.

Čas prejema sporočila je čas, ko elektronsko sporočilo vstopi v IS prejemnika.

- Kraj pošiljanja je kraj, kjer ima pošiljatelj svoj sedež oz. stalno prebivališče v času pošiljanja.
- Kraj prejema je kraj, kjer ima prejemnik svoj sedež oz. stalno prebivališče v času pošiljanja.
- Določbe kraja in časa so dispozitivne – pogodbene stranke jih lahko dogovorno spremenijo (npr. s pogodbo).

ZEPEP overitelji in infrastruktura

- Dejavnost overiteljev lahko opravljajo državni organi, pravne ali fizične osebe.
- Infrastruktura mora zagotavljati
  - upravljanje z javnimi ključi, potrjevanje komplementarnosti ključev, upravljanje pooblastil, objavljanje varnega imenika potrdil, uporabljanje varnih medijev za hranjenje zasebnih ključev, preverjanje identitete končnih uporabnikov, zagotavljanje dokazovanje veljavnosti digitalnega podpisa, zagotavljanje storitev časovnega žigosanja...
  - Sredstva in podatki v zvezi z e-podpisom se morajo hraniti tako dolgo, kot e-podpisani dokumenti.
- Varen časovni žig.
  - Je definiran šele z dopolnitvami ZEPEP-A.
  - Zagotavlja dolgoročno overjanje podpisnika in celovitost e-dokumenta in s tem dolgoročno nezatajljivost prek:
    - dokaza o obstoju e-dokumenta pred določenim časom (v primeru dokazovanja, da je bil dokument oddan pred zakonsko določenim rokom);
    - ohranjanja veljavnosti elektronskega podpisa po poteku veljavnosti digitalnega potrdila;
    - ohranjanja veljavnosti elektronskega podpisa v primeru preklica veljavnosti certifikata (in posledično zasebnega ključa).

Pol pa e-podpisi

- Elektronski podpis so podatki v e-oblikih, ki so vsebovani, dodani ali logično povezani z drugimi podatki in namenjeni preverjanju pristnosti teh podatkov in identifikaciji podpisnika.
- Varen elektronski podpis mora biti izključno povezan s podpisnikom in zasnovan tako, da se podatki, na katere se nanaša, ne morejo neopazno spremeniti. Oblikovan mora biti s sredstvi za varno e-poslovanje in biti overjen s kvalificiranim potrdilom.
- Kvalificirano potrdilo (KP) mora podajati:
  - navedbo, da gre za KP;
  - ime / firmo in državo ali sedež overitelja;
  - ime oz. psevdonim imetnika z navedbo, da gre za psevdonim;
  - dodatne podatke o imetniku potrdila;
  - podatke za preverjanje digitalnega podpisa;
  - začetek in konec veljavnosti potrdila;
  - identifikacijsko oznako potrdila;
  - varen e-podpis overitelja;
  - eventualne omejitve uporabe (npr. transakcijskih vrednosti, kjer se uporablja dotedno potrdilo).
- Zahteve za overitelje:
  - varovanje infrastrukture;
  - zaposleni (najmanj tri osebe z univerzitetno izobrazbo z vsaj 2 letoma del. izkušenj, pravnik);
  - zavarovanje odgovornosti (spočetka 50 M SIT);
  - prijavna služba (pogoji službe za sprejem vlog);
  - notranja pravila (določena z uredbo).
  - nadzor – pristojna inšpekcija (obvezen nadzor) in Agencija za telekomunikacije (prostovoljni nadzor / akreditacija).
- ZEPEP in kazenske določbe - za uporabnika le glede uporabe tujega privatnega ključa, ostalo se vse nanaša na overitelje.

## ZEPEP – nedorečenosti

- Delna funkcionalna ekvivalentnost ročnemu podpisu (traja, preden se izdajo novi sezname preklicanih sporočil (CRL))
- Probleme sinh. Ur
- Problem meta-zaznamkov
- ...

+ občasne pravne nedorečenosti v smislu skladnosti z direktivami EU.

## ZEPEP – A

Dopolnilni zakon, spremeni se odgovornost ponudnikov. Dopolnitev potrebna zaradi EU direktiv.

## Zakon o el. Komunikacijah

- Operaterji morajo hraniti podatke 2 leti od dneva komunikacije, lahko se podaljša za potrebe kazenskega pregona. Ob koncu hranjenja morajo podatke uničiti, razen tistih, za katere je bila izdana odredba za dostop in so bili posredovani pristojnemu organu.

## (kategorije podatkov, ki se hranijo):

- podatki, potrebni za odkritje in prepoznanje vira komunikacije;
- podatki, potrebni za prepoznanje cilja komunikacije;
- podatki, potrebni za ugotovitev datuma, časa in trajanja komunikacije;
- podatki, potrebni za ugotovitev vrste komunikacije;
- podatki, potrebni za razpoznavo komunikacijske opreme uporabnikov.

Kdor grozi, zastrašuje, uničuje, ... ni kul

Ne smeš uporabljati osebnih podatkov v neskladju z namenom

➔ Se kaznuje z denarno kaznijo ali zaporom do enega leta

- Enako se kaznuje, kdor vdre ali nepooblaščeno vstopi v računalniško vodeno zbirko podatkov z namenom, da bi sebi ali komu drugemu pridobil kakšen osebni podatek.

Ne smeš objavljati stvari o žrtvah kaznivih dejanj, kršitev pravic, svoboščin, ... zapor do 3 let

Kraja identitete tud ni kul. Zapor do 3 let. Če je to uradna oseba & zloraba uradnega položaja – zapor do 5 let.

Pregon se začne na predlog.

Ne smeš vdirat v IS-je – zapor do 1 leta.

Neupravičena manipulacija s podatki (+kopiranje) – zapor do 2 leti.

Tudi poskusi so kaznivi. Če rata huda škoda 3meseci-5let

Pri gospodarskem poslovanju neupravičena manipulacija – zapor do 3 let

Velike premoženske koristi – zapor do 5 let

Ne smeš imet, izdelovat, prodajat, dajat, uvažat izvažat ali kako drugače zagotavljati pripomočke za vdor ali neupravičen dostop v inf. sistem.

Pol je pa sam še tak lowkey povzetek