

# MODELI SUVREMENIH POSLOVNIH INFORMACIJSKIH SUSTAVA

---

**Kraljik, Romana**

**Undergraduate thesis / Završni rad**

**2021**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Economics in Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Ekonomski fakultet u Osijeku**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:145:834340>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-05**



*Repository / Repozitorij:*

[EFOS REPOSITORY - Repository of the Faculty of Economics in Osijek](#)



Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Preddiplomski studij (Poslovna informatika)

Romana Kraljik

**MODELI SUVREMENIH POSLOVNIH INFORMACIJSKIH  
SUSTAVA**

Završni rad

Osijek, 2021.

Sveučilište Josipa Jurja Strossmayera u Osijeku  
Ekonomski fakultet u Osijeku  
Preddiplomski studij (Poslovna informatika)

Romana Kraljik

**MODELI SUVREMENIH POSLOVNIH INFORMACIJSKIH  
SUSTAVA**

Završni rad

**Kolegij:** Informatika

**JMBG:** 0010225284

**e-mail:** [rkraljik@efos.hr](mailto:rkraljik@efos.hr)

**Mentor:** prof. dr. sc. Josip Mesarić

**Komentor:** dr. sc. Dario Šebalj

Osijek, 2021.

Josip Juraj Strossmayer University of Osijek

Faculty of Economics in Osijek

Undergraduate Study (Business informatics)

Romana Kraljik

**MODELS OF MODERN BUSINESS INFORMATION  
SYSTEMS**

Final paper

Osijek, 2021.

**IZJAVA**  
**O AKADEMSKOJ ČESTITOSTI,**  
**PRAVU PRIJENOSA INTELEKTUALNOG VLASNIŠTVA,**  
**SUGLASNOSTI ZA OBJAVU U INSTITUCIJSKIM REPOZITORIJIMA**  
**I ISTOVJETNOSTI DIGITALNE I TISKANE VERZIJE RADA**

1. Kojom izjavljujem i svojim potpisom potvrđujem da je završni (navesti vrstu rada: završni / diplomski / specijalistički / doktorski) rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu. Potvrđujem poštivanje nepovredivosti autorstva te točno citiranje radova drugih autora i referiranje na njih.
2. Kojom izjavljujem da je Ekonomski fakultet u Osijeku, bez naknade u vremenski i teritorijalno neograničenom opsegu, nositelj svih prava intelektualnoga vlasništva u odnosu na navedeni rad pod licencom Creative Commons Imenovanje – Nekomercijalno – Dijeli pod istim uvjetima 3.0 Hrvatska.
3. Kojom izjavljujem da sam suglasan/suglasna da se trajno pohrani i objavi moj rad u institucijskom digitalnom repozitoriju Ekonomskoga fakulteta u Osijeku, repozitoriju Sveučilišta Josipa Jurja Strossmayera u Osijeku te javno dostupnom repozitoriju Nacionalne i sveučilišne knjižnice u Zagrebu (u skladu s odredbama Zakona o znanstvenoj djelatnosti i visokom obrazovanju, NN br. 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).
4. izjavljujem da sam autor/autorica predanog rada i da je sadržaj predane elektroničke datoteke u potpunosti istovjetan sa dovršenom tiskanom verzijom rada predanom u svrhu obrane istog.

**Ime i prezime studenta/studentice:** Romana Kraljik

**JMBAG:** 0010225284

**OIB:** 93313766897

**e-mail za kontakt:** romanakraljik@gmail.com

**Naziv studija:** Preddiplomski studij (Poslovna informatika)

**Naslov rada:** Modeli suvremenih poslovnih informacijskih sustava

**Mentor/mentorica rada:** prof. dr. sc. Josip Mesarić

U Osijeku, 7. 9. 2021. godine

Potpis Romana Kraljik

# Modeli suvremenih poslovnih informacijskih sustava

## SAŽETAK

Tema ovog završnog rada je *Modeli suvremenih poslovnih informacijskih sustava*. Cilj rada je dati pregled ključnih modela kojima se opisuju informacijski sustavi te pokazati način funkcioniranja suvremenog modela temeljenog na grafičkom jeziku za modeliranje informacijskih sustava koji nosi naziv UML (engl. Unified Modeling Language). Tipični statički i dinamički modeli primijenit će se na prikazu nekoliko primjera jednostavnog IS-a. U radu su se koristile opće metode istraživanja (analiza i sinteza). U radu su se osim općih metoda koristile i induktivno-deduktivne metode te metode kompilacije. Metodama grafičke reprezentacije koristilo se u opisu različitih aspekata sustava pomoću UML-a. U uvodnom dijelu rada opisani su informacijski sustavi, poslovni sustavi i njihova međusobna povezanost. Njihova svrha i podjela navedena je u daljnjem razvoju završnog rada. U nastavku su navedeni aspekti modeliranja, te opis pojmova strukturnih i objektno orijentiranih modela. Prije opisivanja vrsta metoda, navedene su definicije metoda i metodologije. Vrste statičkih i dinamičkih modela detaljno su opisane, a nekoliko njih iz svake vrste potkrijepljene su primjerom, odnosno grafički su prikazane te su napravljene uz pomoć programa Astah UML. U raspravi završnog rada navedeni su koristi i nedostaci pojedinih modela. U zaključku samog završnog rada navedene su završne misli po pitanju modela te su dane zaključne sugestije za alternativne modele ili njihove kombinacije.

Ključne riječi: informacijski sustavi, poslovni sustavi, modeli, dijagrami, UML

# **Models of modern business information systems**

## **ABSTRACT**

The topic of this final paper are models of modern business information systems. The aim of this paper is to give overview of key models that describe information systems and to show how a modern model based on a graphical language for modeling information systems called UML (Unified Modeling Language) works. Typical static and dynamic models will be applied to display several examples of simple IS. Methods that were used in this paper are general research methods (analysis and synthesis). In addition to general methods, in this paper are also used inductive-deductive methods and compilation methods. Graphical representation methods have been used to describe various aspects of the system using UML. In the introductory part of the paper were described information systems, business systems and their interconnection. Their purpose and division is stated in the further development of the final paper. Next are aspects of modeling and a description of the concepts of structural and object-oriented models. Before describing the types of methods, there are given definitions of methods and methodologies. The types of static and dynamic models are described in detail and several of them of each type are supported by an example and made with the help of Astah UML program. In the discussion of the final paper, there is a list of the advantages and disadvantages of each individual model. In the conclusion of the final paper, there are given final thoughts on the model and the final suggestions for alternative models or their combinations.

Keywords: information system, bussines system, models, diagrams, UML

# SADRŽAJ

<b>1. Uvod</b> .....	1
<b>2. Metodologija rada</b> .....	2
<b>3. Informacijski sustavi (IS)</b> .....	2
<b>3.1. Struktura i funkcija sustava</b> .....	5
<b>3.2. Povezivanje IS i PS</b> .....	7
<b>4. Modeli informacijskog sustava</b> .....	8
<b>4.1. Aspekti modeliranja</b> .....	12
<b>4.2. Strukturni i objektno orijentirani modeli</b> .....	13
<b>4.3. UML jezik za modeliranje IS-a</b> .....	13
4.3.1. Aspekti modeliranja .....	16
4.3.2. Statički modeli i primjeri upotrebe.....	17
4.3.3. Dinamički modeli i primjeri upotrebe .....	21
<b>5. Rasprava</b> .....	26
<b>6. Zaključak</b> .....	28
<b>Literatura</b> .....	29
Popis slika i tablica.....	31



## 1. Uvod

S razvojem tehnologije, organizacijama je postalo puno lakše u kreiranju sustava koji će koristiti u svom poslovanju. Na raspolaganju im stoje brojne mogućnosti kako bi to i uspjele ostvariti. Također, s razvojem tehnologije postalo je i sve lakše doći do željenih informacija kojih svakim danom ima sve više i više. Njihova raznovrsnost i broj ne pomaže samo organizacijama, poduzećima ili društvima kako bi došli do željenih informacija koji su im potrebni već i običnim pojedincima.

Informacijski sustavi, ovisno o njihovoj svrsi služe, manje ili više složene strukture ljudi, organizacija, strojeva (računala), programskih sustava, podataka i informacija, koji se zbog svoje službenosti prikazuju različitim modelima. Statičke strukture opisuju se različitim modelima, koji daju prikaz onoga što sustav jest. U funkcioniranju sustava opis se mora napraviti upotrebom dinamičkih modela, koji osiguravaju uvid u to kako sustav radi. Modeli se koriste u različitim fazama razvoja informacijskog sustava, u opisu struktura i načinu funkcioniranja.

Općenito su modeli pojednostavljeni opisi realnosti (sustava) koji se koriste sa svrhom da se opišu ključne značajke sustava s danog aspekta. S razvojem složenosti sustava u upotrebu su uvedeni različiti novi modeli s namjerom da sustav opišu s potrebnog aspekta.

Walter Wriston, čelni čovjek Citicorpa, je još sedamdesetih godina dvadesetog stoljeća primijetio kako je informacija o novcu mnogo važnija nego sam novac. (Srića, Spremić, 2000, str. 4)

Cilj rada je dati pregled ključnih modela kojima se opisuju informacijski sustavi te pokazati način funkcioniranja suvremenog modela temeljenog na grafičkom jeziku za modeliranje informacijskih sustava koji nosi naziv UML (engl. Unified Modeling Language).

## 2. Metodologija rada

U radu će se koristiti opće metode istraživanja (analiza i sinteza) kao i induktivno-deduktivne metode te metoda kompilacije. Posebne specifične metode obuhvatit će sistemski pristup koji će se koristiti u prikazu klasičnih metoda dok će se metodom grafičke reprezentacije pomoću UML-a koristiti u opisu različitih aspekata rada.

## 3. Informacijski sustavi (IS)

Pod pojmom informacijski sustav podrazumijeva se onaj sustav koji pomaže u prikupljanju, pronalaženju, obradi, pohrani, skladištenju i isporuci informacija i podataka radi donošenja odluka.

Postoje brojne definicije informacijskih sustava. U nastavku se navode neke od definicija koje ukazuju na strukturu i funkciju informacijskog sustava.

„Informacijski sustav jest sustav koji prikuplja, pohranjuje, čuva, obrađuje i isporučuje informacije važne za organizaciju i društvo, tako da budu dostupne i upotrebljive svakome kome su potrebne. Informacijski sustav aktivni je društveni sustav koje se može, ali ne mora, koristiti suvremenom informacijskom tehnologijom“. (Varga, Strugar, 2016, str. 6)

Informacijski sustavi sastoje se od sljedećih osnovnih dijelova: (Srića, Spremić, 2000, str. 8-9)

- Hardware – predstavlja fizički dio samog informacijskog sustava, a pod time podrazumijevamo osobna računala, različitu mrežnu opremu, usmjerivače i sl.
- Software – je tzv. nevidljivi dio informacijskog sustava, ono dolazi u obliku programskog rješenja, odnosno algoritma koji pokreće hardver
- Dataware – to su svi podatkovni resursi, tj. metode i načini organizacije baza i skladišta podataka
- Lifeware – sve osobe koje se služe informacijskim sustavom
- Netware – mrežna i komunikacijska rješenja koja spajaju sve elemente u jednu cjelinu
- Orgware – svi organizacijski postupci i metode koji spajaju sve navedene elemente u jednu zajedničku cjelinu

Uz pojam informacijski sustavi (IS) vežu se dva načela: ontološko i teleološko načelo.

„Ontološko načelo proizlazi iz datosti proučavanog fenomena, njegovih koncepata i odnosa među njima.“ (Mesarić, Šebalj, 2020.)

Pod teleološkim se načelom razmatra informacijski sustav s aspekata svrhe koju predstavlja u danom kontekstu. Klasični kontekst može uključivati državne ustanove, poslovne sustave, neprofitne organizacije poput škola, biblioteka i sličnih ili može uključivati tehnološki proces ili neki drugačiji ograničeni kontekst u organizaciji. (Mesarić, Šebalj, 2020).

Informacijski sustavi (IS) se dijele na nekoliko osnovnih vrsta (Srića, V. i Spremić, M., 2000, str. 9):

- klasični ili transakcijski informacijski sustavi
- ekspertni sustavi
- sustavi za potporu odlučivanju

Klasični ili transakcijski IS kao osnovni dio svakog informacijskog sustava pripada operativnoj razini poslovanja, gdje pruža potporu transakcijama i tekućim procesima. Izvještajni sustav klasičnog IS pruža još neobrađene informacije o tijeku neke aktivnosti ili procesa.

Ekspertni sustavi posjeduju mogućnost pohranjivanja stručnog znanja, tj. znanja stručnjaka iz pojedinog područja te koriste njihovo znanje pri odlučivanju, donošenju odluka i stvaranju novog znanja.

Osim općih informacija i podataka koje pružaju, sustavi za potporu odlučivanju posjeduju baze modela, statistika, dokumenata i prognoza koji pomažu menadžerima u procesima odlučivanja.

Prema (Gupta, 2000, str. 17), informacije koje se koriste moraju imati sljedeće navedene karakteristike:

- točnost – informacije ne smiju sadržavati pogreške,
- pravovremenost – informacije trebaju biti isporučene na vrijeme,
- kompletnost – informacije trebaju biti potpune,
- pouzdanost – informacije ne smiju imati kontradiktorni značaj za donositelja odluke.

Prema (Čerić, V. i Varga, M., 2004, str. 46) postoji šest faza izgradnje informacijskog sustava:

- 1. Planiranje informacijskog sustava:** Drugi naziv za ovu fazu je i izrada strategije informacijskog sustava. Ovu fazu nije potrebno obavljati za manje informacijske sustave, ali kod velikih informacijskih sustava može biti ključna za cijeli projekt razvoja jer na temelju rezultata dobivenih u ovoj fazi, poslovodstvo donosi odluke o daljnjem razvoju projekta. Prije izvršavanja ove faze potrebno je ustanoviti bitne aspekte: potrebno je identificirati nedostatke postojećeg sustava, procijeniti ciljeve novog informacijskog sustava te definirati korisnike sustava. „Faza završava ocjenom izvedivosti informacijskog sustava i razrađenim planom daljnjeg razvoja kroz sljedeće faze, procjenom potrebnih financijskih sredstava i drugih resursa, procjenom mogućih rizika daljnje izgradnje itd.“
- 2. Analiza poslovnog sustava:** Uloga analize poslovnog sustava je odrediti informacijske zahtjeve ili korisničke zahtjeve, odnosno što treba informacijski sustav raditi kako bi ispunio ono što korisnici očekuju od njega. Analitičari se koriste različitim vještinama i tehnikama, kao npr. rad na radnim sjednicama, intervjuiranje korisnika, „oluja mozgova“ (engl. brainstorming), proučavanje dokumenata ili proučavanje izvršavanja poslovnih procesa. Utvrđivanjem poslovnih procesa, objekata i događaja utvrđeni su korisnički zahtjevi.
- 3. Oblikovanje informacijskog sustava:** u ovoj fazi se razmatra kako će informacijski sustav raditi. Sama struktura podataka informacijskog sustava je prikazana dijagramima entiteta i veza koja se oblikuje u bazu podataka koja će biti smještena na računalo. Također, u ovoj fazi se određuju i tehnološki te organizacijski uvjeti za rad sustava, kao što su potrebno računalo i druga tehnička oprema, potrebni programi, te organizacijski i kadrovski preduvjeti za rad informacijskog sustava.
- 4. Izrada informacijskog sustava:** U fazi izrade informacijskog sustava se prethodno definirani procesi programiraju te time dobivaju računalni programi. Programiranje se može izvršavati uz pomoć različitih programskih jezika, koji se po namjeni razlikuju jedan od drugog. Klasični su proceduralni programski jezici poput COBOL ili C, specijalizirani je npr. SQL koji radi s bazom podataka, te moderniji objektni jezici kao C++ ili Java. Faza izrade informacijskog sustava završava izradom, provjerom i dokumentiranje programskog dijela informacijskog sustava.

- 5. Uvođenje novog sustava u rad:** U ovoj je fazi potrebno staviti novi informacijski sustav u funkciju. Korisnici će tijekom pripreme za rad i izobrazbe s novim informacijskim sustavom imati mogućnost se upoznati sa svrhom informacijskog sustava, u detalje naučiti nove procedure obavljanje poslova te kako koristiti računalnu opremu na radnim mjestima. Završni korak uvođenja novog sustava jest detaljna provjera cijelog sustava, odnosno testiranje sustava. Ako je sustav složen, postoje tri načina kako ga uvesti, a to su: izravno uvođenje novog sustava, paralelno uvođenje novog sustava i rad starog, te postupno uvođenje dijela novog sustava.
- 6. Održavanje informacijskog sustava:** U ovoj fazi, sustav je u svakodnevnom korištenju. Svaki je informacijski sustav potrebno održavati, a razlog je taj kako bi se ispravile uočene pogreške i nedostaci koji su otkriveni tijekom rada sustava, te kako bi se sustav mogao prilagoditi promjenama poslovnog sustava koji su nastali nakon njegova uvođenja. Ako se sustav slabo održava ili ne održava, sve će lošije ispunjavati zahtjeve korisnika. U tom će se slučaju postaviti zahtjev za uvođenjem novog informacijskog sustava, a to znači kako je potrebno napraviti novi informacijski sustav koji će prolaziti sve navedene faze izgradnje informacijskog sustava.

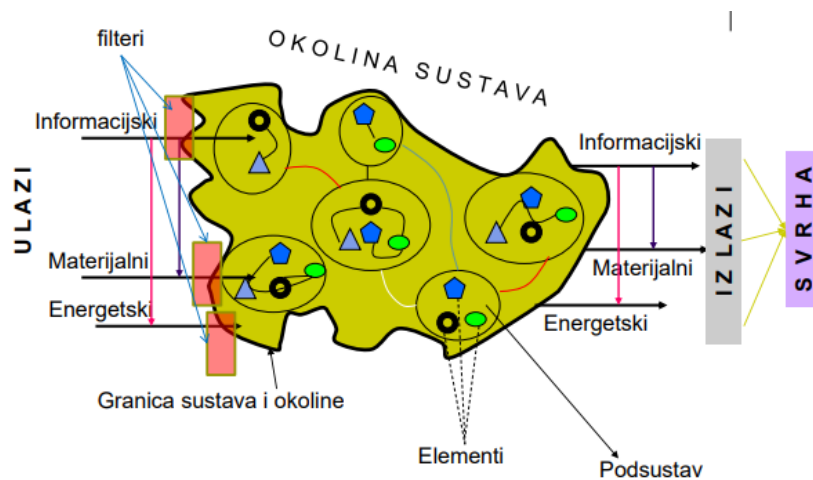
„Cilj je informacijskog sustava opskrbiti poslovni sustav informacijama potrebnima izvršnom podsustavu za izvođenje poslovnog procesa, upravljačkom podsustavu za upravljanje poslovnim sustavom, te pri suradnji i komunikaciji unutar poslovnog sustava i prema okolini.“ (Čerić, Varga, 2004, str. 20)

Prema istom izvoru, „informacijski sustav može se podijeliti na više dijelova, od kojih je svaki zadužen za izvršenje jednog od prethodno opisanih ciljeva. Sustav za obradu transakcija (operativni sustav) služi za izvođenje poslovnog procesa, sustav za potporu odlučivanju (informativni sustav, analitički sustav) služi za upravljanje poslovnim sustavom, te sustav za komunikaciju, suradnju i individualni rad (uredski sustav).“

### **3.1. Struktura i funkcija sustava**

„Sustav je zbir dijelova koji funkcioniraju zajedno kao cjelina i ostvaruju neki zajednički cilj. Pri tome sustav ne čini bilo koji zbir dijelova, već samo takav zbir dijelova koji se ne mogu rastaviti, a da se ne izgube svojstva sustava“ (Varga, Strugar, 2016, str. 1).

Može se zaključiti kako se pod pojmom sustava podrazumijeva skup raznovrsnih komponenti ili elemenata koje zajednički doprinose proizvodnji nekog rezultata, koji ne može postići njegova komponenta ili element samostalno.



Slika 1. Konceptualni prikaz sustava

Izvor: Mesarić, Šebalj (2019.), raspoloživo na:

<http://www.efos.unios.hr/oblikovanje-implementacija-is/predavanja/?lang=hr>

Pristupljeno: 14.6.2021.

Strukturu sustava čine njegove komponente, položaj jedne komponente u odnosu na drugu komponentu, te odnos komponenata kao cjeline i povezanost, odnosno veze među njima. Među pojedinim komponentama sustava može se uspostaviti veza preko drugih komponenti i to neposredno ili posredno.

„Pod funkcijom sustava podrazumijevamo svrhu postojanja sustava, ulogu koju sustav ima u svojoj okolini i način ostvarivanja svrhe. Funkcija sustava proizlazi iz stajališta promatranja. Iz jasno definirane funkcije sustava utvrđuju se njegove komponente i njihovi međudnosi (struktura). (Pavlić“ prema (Mesarić, Šebalj, 2019).

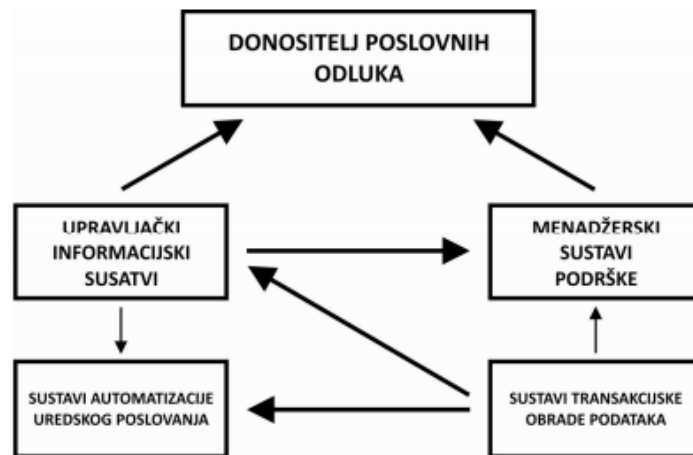
Prema (Velimir Srića, 1981, str. 15), postoji deset zapovijedi teorije sustava:

1. „Naglasak na uzajamnoj povezanosti i zavisnosti elemenata, atributa, događaja, predmeta ili pojava koje čine sustav.
2. Elementi cjeline (sustava) ne promatraju se zasebno, nego u okviru procesa funkcioniranja čitave cjeline, što nazivamo holističkim pristupom.

3. Elementi sustava u uzajamnoj interakciji orijentirani su postizanju ciljeva sustava, što nazivamo teleološki princip ili svrhovitost.
4. Sustav je u interakciji sa svojom okolinom kao fenomen materije, energije i informacija, što ćemo nazvati ulaznim veličinama sustava.
5. Proces ili funkcija sustava izražava se kao transformiranje ulaznih veličina u izlaz, odnosno  $I = f(U)$ , ili  $I=T \cdot U$ .
6. Entropija, kao pojam suprotan organiziranosti i informaciji, predstavlja mjeru nereda, nefunkcionalnosti i neorganiziranosti sustava.
7. Sustav postiže ciljeve procesima regulacije, na osnovu povratne veze – usporedba izlaznih i ciljnih veličina.
8. Svaki sustav je dio nekog većeg sustava.
9. Elementi sustava ili podsustava diferenciraju se i specijaliziraju za pojedine funkcije.
10. Isto stanje sustava u jednom vremenskom diskretnom trenutku je moguće ostvariti na različite načine.“

### **3.2. Povezivanje IS i PS**

U svakoj je organizaciji potrebno upravljati određenim sustavom kako bi ona mogla donijeti ispravne odluke i ostvariti svoje zadane ciljeve. Za donošenje pravih i smislenih odluka, organizacija mora posjedovati bitne informacije, a te informacije joj pruža informacijski sustav. Kao što je na navedeno u prethodnim poglavljima, informacijski sustav je sustav koji služi za prikupljanje podataka ili informacija, koji su potrebni kako bi pomogle onima kojima su potrebne u donošenju odluka. Kako bi poslovni sustav bio uspješan, mora posjedovati odgovarajući informacijski sustav, koji je podržan najsuvremenijom tehnologijom. Informacijski sustav prikuplja podatke iz raznih izvora, bili oni unutarnji ili vanjski. Na kraju, poslovni sustav koristi te podatke prikupljene od strane informacijskog sustava.



Slika 2. Prikaz sheme poslovnog informacijskog sustava

Izvor: Nastavni materijali za kolegij „Poslovni informacijski sustavi“, 2013.

<http://www.efos.unios.hr/poslovni-informacijski-sustavi/wp-content/uploads/sites/216/2013/04/1.-POSLOVNI-INFORMACIJSKI-SUSTAVI.pdf>

Pristupljeno: 14.6.2021.

U svakom poslovnom sustavu prikupljaju se, raspodjeljuju, obrađuju, isporučuju i čuvaju različite informacije i podaci bitni za organizaciju. Neki oblik informacijskog sustava, bio on formalan ili neformalan, postoji u svakom poslovnom sustavu.

„IS nije usmjeren na određeni problem nego svakodnevno i stalno prati i nadzire promjene u okruženju tako da tvrtka može prilagođavati svoje poslovanje i brzo reagirati na promjene na tržištu“ (Srića, Spremić, 2000., str. 10).

#### 4. Modeli informacijskog sustava

U informacijskom sustavu je gotovo nemoguće opisati sve njegove detalje jer je informacijski sustav kao cjelina previše složen.

„Modeliranje se može definirati kao čin predstavljanja nečega, obično u manjem obujmu ili sa manje detalja. Uz upotrebu alata za modeliranje (npr. poslovnih procesa) modeliranje se može shvatiti kao čin grafičke reprezentacije poslovnih procesa ili softvera. Model tako kreiran može se koristiti za određene aspekte sustava koji se predstavlja modelom (podaci,



dokumenti, komunikacija). Studija modela omogućava uvid u razumijevanje modeliranog sustava.“ (Mesarić, Šebalj, 2020/2021).

Model je moguće definirati kao „jednostavnu sliku stvarnosti u kojoj dolazi do izražaja najznačajnija svojstva te stvarnosti. Konstrukciju modela, odnosno opis stvarnosti se može izvršiti različitim „jezicima“ (izražajnim sredstvima). Kod opisivanja stvarnosti, nužno je postojanje izomorfije, odnosno, svojstva iz modela bi se trebala jednoznačno preslikati u svojstva iz stvarnosti.“ (Mesarić, Šebalj, 2020).

Prema prethodno navedenom izvoru, kod modeliranja IS-a postoji nekoliko vrsta modela:

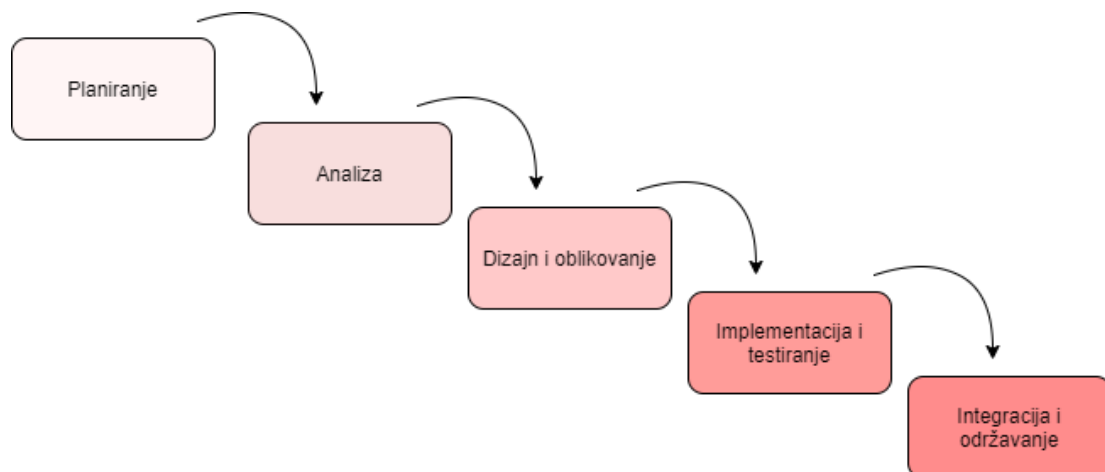
- Modeli za pristup razvoju informacijskih sustava
- Modeli strukturnih elemenata i strukturne analize
- Konceptualni model
- Modeli aspekata s kojih se promatra IS
- Integrativni model
- Metamodeli navedenih modela i modelskih podsustava

Svaki pojedini model određuju pojedine sastavnice. Tako model za pristup razvoju informacijskih sustava određuje faza u izgradnji sustava i dinamika izgradnje. Model strukturnih elemenata i strukturne analize određuje dvije strukture, a to su: strukture sustava s različitih aspekata i strukturalne odnose. Konceptualni model određuje obuhvat i domenu (model ciljeva i funkcija, sistemski model, kontekstni model, model sudionika i njihove uloge). Modeli aspekata s kojih se promatra IS određuje nekoliko modela: model podataka, model događaja, model resursa, model programa te model funkcija i procesa. Integrativni modeli obuhvaćaju konceptualne modele u koji ulaze strukturalni modeli i objektno orijentirani modeli temeljeni na UML notifikaciji te modeli strukturnih elemenata.

„Također, kod odabira modela, moraju se koristiti određene tehnike i alate za njegov opis poput pseudo-koda kao pseudojezika, jezične strukture prirodnog jezika te grafički prikazi dogovorenim (standardiziranim) skupom simbola i notacija za pojedine aspekte sustava.“ (Mesarić, Šebalj, 2020).

Vodopadni (waterfall) pristup razvoja informacijskog sustava je pristup zasnovan na slijednom napredovanju iz faze u fazu koje izgledom podsjeća na slap (engl. waterfall). Specifičnost vodopadnog pristupa jest ta što se ne može krenuti u novu fazu razvoja sve dok se ne završi prethodna razina. Što znači da su zabranjene, odnosno ne postoji mogućnost

vraćanja i ispravljanja rezultata u prethodnim razinama. Ovaj pristup ili model je prikladniji za velike projekte, odnosno investicije. Svaka se faza dokumentira te je sam pristup izrazito temeljit. Ideja je da je bolje pogriješiti u ranijim fazama i ranije uočiti pogrešku i ispraviti ju, nego u kasnijim fazama. Nedostatci vodopadnog pristupa su ti da je vremenski zahtjevan, ne postoji mogućnost predviđanja budućih događaja niti mogućnost predviđanja troškova vezanih za pojedine faze, procjena rizika ne postoji, a sam sustav je upotrebljiv tek kada je cijeli model gotov u potpunosti. Model mora biti analiziran od strane visoko stručnih osoba. „Vodopadni pristup posjeduje logičko i fizičko oblikovanje. U logičko oblikovanje ulazi izrada odgovarajućeg modela, oblikovanje programskih struktura i slično, dok pod fizičko oblikovanje ulazi fizičko oblikovanje baze, dodavanje „fizičkih“ elemenata programima te oblikovanje korisničkog sučelja.“ (Mesarić, Šebalj, 2020).



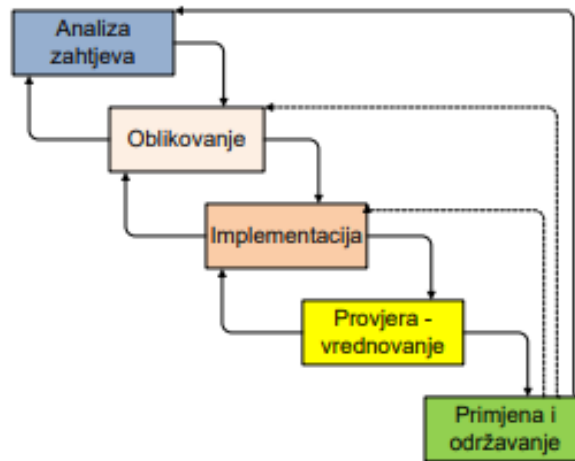
Slika 3. Primjer vodopadnog (engl. waterfall) pristupa

Izvor: TheRespawned, 2018.

<https://www.pinterest.com/pin/715298353304934033/>

Pristupljeno: 14.6.2021.

Modificirani vodopadni pristup je pristup koji je po razinama sličan klasičnom vodopadnom pristupu, osim što modificirani vodopadni pristup ima mogućnost povratne veze, odnosno mogućnost povratka u prethodne razine i promjene rezultata u njima. Tako se istovremeno u više faza mogu odvijati promjene. Ovaj model je adekvatniji kada se ne zna unaprijed konačan izgled sustava.



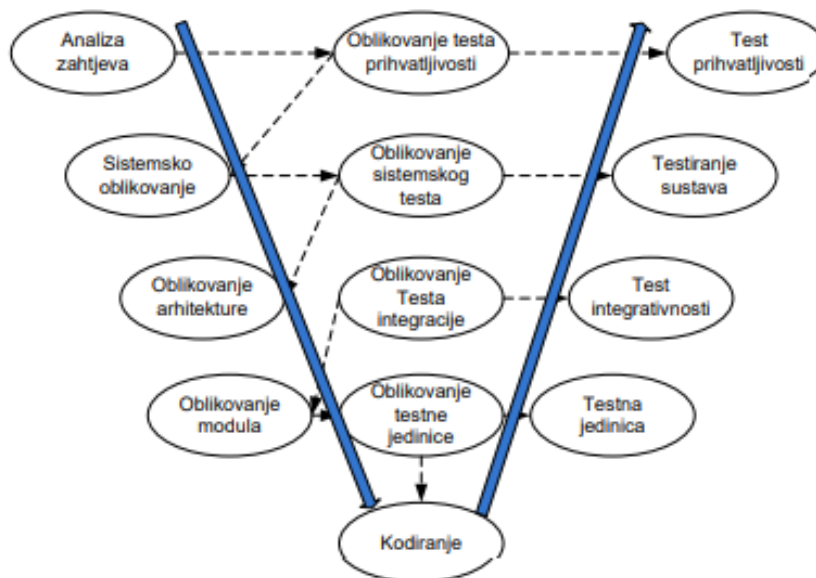
Slika 4. Primjer modificiranog vodopadnog pristupa

Izvor: Mesarić, Šebalj (2019.), raspoloživo na:

<http://www.efos.unios.hr/oblikovanje-implementacija-is/predavanja/?lang=hr>

Pristupljeno : 14.6.2021.

V – model kako mu i ime govori podsjeća na slovo V koji je stvoren evolucijom softverskog testiranja. Model je stvoren prema činjenici kako je sustav stvoren od dijelova, koji se i dalje mogu spajati u druge dijelove i tako stvoriti određenu hijerarhiju. Baza ovog modela pokazuje kako se nakon svake pojedine faze provodi testiranje i ispravljanje pogrešaka.



Slika 5. Primjer V – modela

Izvor: Mesarić, Šebalj (2019.), raspoloživo na:

<http://www.efos.unios.hr/oblikovanje-implementacija-is/predavanja/?lang=hr>

Pristupljeno: 14.6.2021.

Postoje i druge vrste modela poput spiralnog (Boehm-ov) modela, evolucijskog modela, inkrementalnog modela i dr.

#### 4.1. Aspekti modeliranja

Za pristup izgradnji IS-a potrebno je izabrati određene modele i metodologije. Veličina i obujam projekta jedna je od glavnih stavki koje se trebaju gledati ako se želi odabrati ispravan model i metodologija. Ciljevi koji se žele postići ili problemi koji se žele riješiti uvođenjem informacijskog sustava i raspoloživo vrijeme su također bitni aspekti o kojima ovisi odabir modela. Ono također ovisi i o resursima koji stoje organizaciji na raspolaganju, modalitetima izgradnje te znanje kojim se raspolaže o navedenom problemu zbog kojeg se razvija IS.

Prilikom izgradnje informacijskog sustava, potrebno je oblikovati određene elemente arhitekture: (Mesarić, 2015/2016)

1. Poslovna arhitektura – u ovu arhitekturu ulaze organizacija podataka, procesa i funkcija, razdvajanje različitih zahtjeva, povezivanje domena...
2. Razvojna arhitektura – u ovu arhitekturu se vežu razna ograničenja (lokacije, ljudi...), integrativna komponenta ove arhitekture jesu protokoli, u ovu arhitekturu ulaze svi poslovni procesi koji su uključeni u nju
3. Logička arhitektura –u ovu arhitekturu ulaze svi funkcionalni zahtjevi poslovnih procesa koji će biti implementirani kasnije u poslovni proces
4. Operativna arhitektura – u nju su uključeni svi rizici i sustavi zaštite, definiraju se prava pristupa logičkih grupa korisnika
5. Fizička arhitektura – koji će se operacijski sustavi, primjenski softveri ili sučelja koristiti.

#### **4.2. Strukturni i objektno orijentirani modeli**

Strukturirane i objektno orijentirane ili usmjerene metodologije ulaze u vrste komercijalne metodologije. Svaka od dviju navedenih metodologija ima ključne probleme. Kod strukturnih metoda kao problem ističu se nefleksibilnosti i vremenske zahtjevnosti. Strukturne metode nisu uspjele riješiti problem dinamike sustava te nisu napravljene za brzo promjenjive podatke.

„Objektne metodologije opisuju sustav sa svih aspekata uključujući njegovu dinamiku, zahtijevaju visoku razinu apstrakcije, fleksibilnost i adaptabilnost definiranih klasa na različite tipove problema.,, (Mesarić, 2020)

#### **4.3. UML jezik za modeliranje IS-a**

UML je skraćenica koja stoji iza punog naziva Unified Modeling Language. UML predstavlja moderan pristup modeliranja i jednostavan jezik pomoću kojeg se vizualno prikazuje model sustava radi stvaranja i razvoja softvera ili sustava.

Prema (Miles, Hamilton, 2006, str. 23) jezik za modeliranje može biti pseudo-kod, stvarni kod, slike, dijagrami, opisi, odnosno sve ono što može pomoći u opisivanju nekog sustava.

Elementi koji čine jezik za modeliranje nazivaju se notacijama, a ono što te notacije predstavljaju, odnosno znače naziva se semantika.

Kod svakog pristupanja određenom jeziku za modeliranje, postoje određene prednosti i nedostaci.

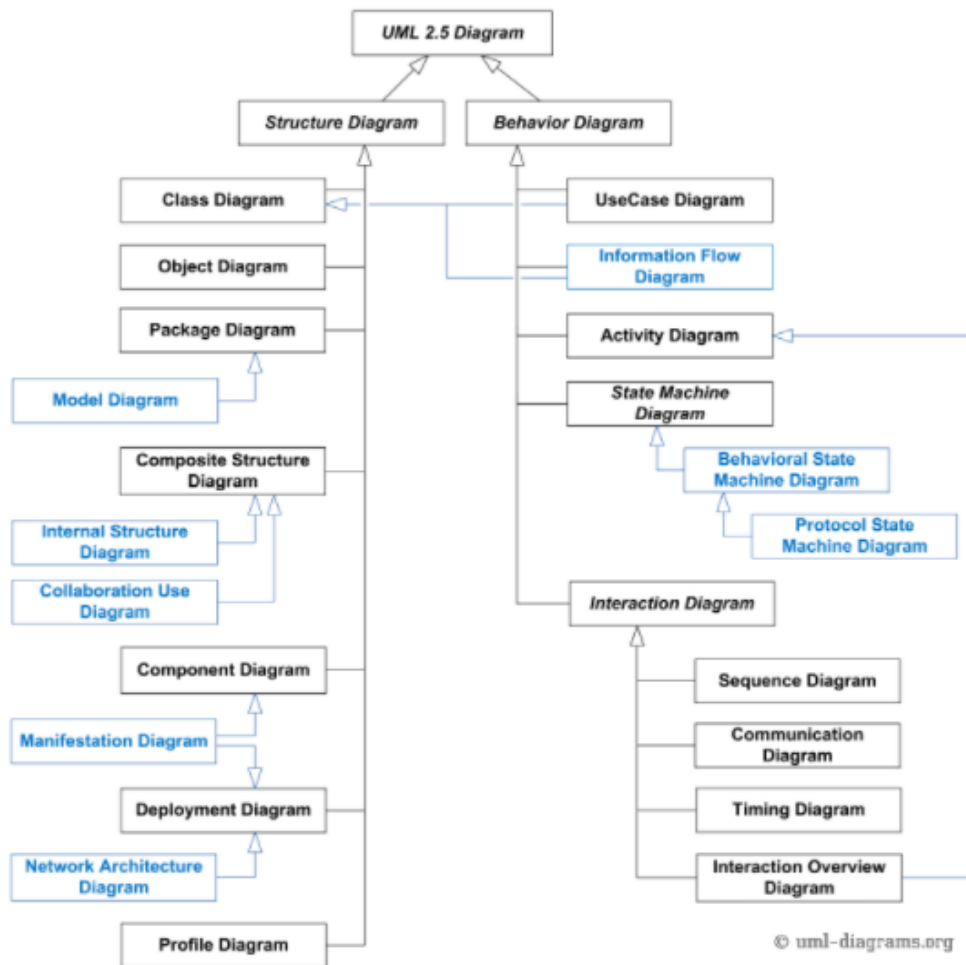
Tko i UML ima šest glavnih prednosti (Hamilton, Miles, 2006, str. 24):

- Formalni je jezik (svaki je element ima strogo definirano značenje)
- Sažet je (cijeli jezik se sastoji od jednostavnih i neposrednih notacija)
- Cjelovit (opisuje sve bitne aspekte sustava)
- Skalabilan (jezik je dovoljno razvijen za upravljanje masivnim sustavnim modeliranjem projektima, ali je ujedno i pogodan za manje projekte)
- Građen na lekcijama (UML je najbolja praksa u objektno orijentiranoj zajednici tijekom posljednjih 15 godina)
- Standardan (građen i kontroliran prema svjetskim standardima).

Prema (Miles, Hamilton, 2006:34), navodi se kako Martin Fowler opisuje tri načina na koji ljudi najčešće koriste UML, a to je kao skicu, kao nacrt, te kao programski jezik.

Prva verzija UML-a je izašla u prosincu 1997. godine. Te godine je UML prihvaćen kao standard od strane OMG-a (Object Management Group). Trenutačna verzija UML-a je 2.5. koja je izašla u svibnju 2015. godine.

UML 2.5. može se podijeliti na dvije glave kategorije dijagrama: strukturne dijagrame i dijagrame ponašanja. Svaka od dviju kategorija dijeli se na više različitih vrsta:



Slika 6. Prikaz kategorija dijagrama 2.5. UML verzije

Izvor: Kirill Fakhroutdinov, 2013.

<https://www.uml-diagrams.org/uml-25-diagrams.html>

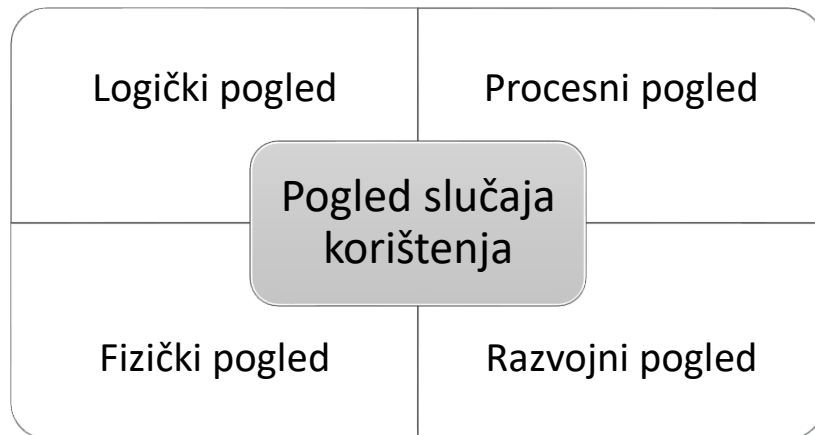
Pristupljeno: 14.6.2021.

Iz prethodne slike može se vidjeti cjelokupna struktura dijagrama iz obje navedene kategorije. Važno je napomenuti kako oni dijagrami koji su označeni svjetlo plavom bojom ne ulaze u službenu taksonomiju UML 2.5. dijagrama.

Dijagrami UML-a mogu se podijeliti na još dvije vrste modela: dinamičke i statične. Njihova podjela navesti će se u posebnim poglavljima namijenjenim isključivo ovim dvjema vrstama.

### 4.3.1. Aspekti modeliranja

Za svaki aspekt potrebno je dati statički i dinamički opis sustava. Krunchetenov 4+1 model razbija model na cjeline, tako da svaka cjelina obuhvaća specifični aspekt sustava.



*Slika 7. Krunchetenov 4+1 model*

Izvor: vlastita izrada prema Miles, Hamilton (2006.)

Pristupljeno: 14.6.2021.

Logički pogled opisuje dijelove sustava. Pokazuje od čega se sastoji sustav te kako određeni dijelovi sustava surađuju međusobno. Procesni pogled prikazuje procese koji se događaju unutar sustava. Razvojni pogled prikazuje module i komponente u koje su organizirani dijelovi sustava. Fizički pogled opisuje kako dizajn sustava poprima entitete iz stvarnog svijeta, koji su opisani u prethodna 3 pogleda. Dijagrami fizičkog pogleda pokazuju kako se apstraktni dijelovi preslikavaju u konačno razvijen sustav. Pogled slučajeva korištenja prikazuje kako sustav treba raditi i o njemu ovisi i oslanjanju se prethodna četiri pogleda. Ovaj pogled prikazuje kako sustav funkcionira izvana. Slučaj korištenja kreira se u najranijim fazama oblikovanja i to najčešće kao prvi dijagram. On daje prikaz vrijednosti, koju sam sustav pruža korisnicima. (Hamilton, K., Miles, R., 2006:36, prevela autorica)



#### 4.3.2. Statički modeli i primjeri upotrebe

Prema (Mesarić, 2015/2016), statistički modeli prikazuju, tj. opisuju potrebne entitete, odnosno sve stavke o kojima se vode podaci i svojstva tih entiteta

Statički model obilježavaju sljedeći dijagrami:

- Dijagram slučajeva korištenja (engl. UseCase Diagram)
- Dijagram klasa (engl. Class Diagram)
- Dijagram objekta (engl. Object Diagram)
- Dijagram komponenti (engl. Component Diagram)
- Dijagram rasporeda (engl. Deployment Diagram).

**Dijagram slučajeva korištenja** mora omogućiti definiranje korisničkih zahtjeva koje sustav mora ispuniti. Ovaj dijagram opisuje koji korisnik koristi koje funkcionalnosti sustava, ali ne definira detalje implementacije, odnosno ne bavi se detaljima provedbe. Ovaj dijagram prikazuje ponašanje sustava točno onako kako ga mora vidjeti korisnik.

Kako bi dijagram izvršio svoju zadaću i radio onako za što je zamišljen, potrebno je prvo odgovoriti na sljedeća pitanja (Seidl, 2012:23):

1. Što se opisuje? (Sustav)
2. Tko sudjeluje u interakciji sa sustavom? (Sudionici)
3. Što sudionici mogu raditi? (Slučajevi korištenja).


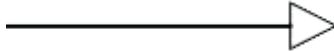
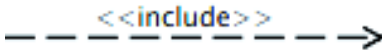
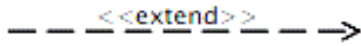
Prilikom započinjanja kreiranja dijagrama, potrebno je definirati zahtjeve sustava. Kao prvi korak može se navesti određivanje sudionika, kao osoba koje su u interakciji sa sustavom. Sudionik je vanjski entitet koji nije direktno dio sustava, ali je povezan s njime. Sudionike je potrebno pisati kao imenicu u jednini i ne bi smjeli imati imena koja su povezana s organizacijom poduzeća. Ono može biti stvarna osoba (npr. student, kupac...) ili neki drugi sustav, na primjer neka aplikacija.

Drugi korak jesu slučajevi korištenja, odnosno potrebno je pronaći situacije (slučajeve) gdje će se taj sustav koristiti kako bi mogao izvršavati određene zadatke sudionika. Slučaj

korištenja je apstraktni zadatak kojeg izvode sudionici i potrebno ga je pisati kao glagol, odnosno neka konkretna radnja radi jednostavnosti.

Kako bi se lakše uočilo slučajeve korištenja, potrebno je stvoriti određeni scenarij. Scenarij predstavlja korake koji slijedno opisuju međusobnu interakciju između sustava i sudionika.

Kod korištenja dijagrama slučajeve korištenja, ali i kod nekih drugih dijagrama, postoje različite vrste veza.

VRSTA VEZE	PRIKAZ
Asocijacija	
Generalizacija	
Uključivanje (engl. include)	
Proširenje (engl. extend)	

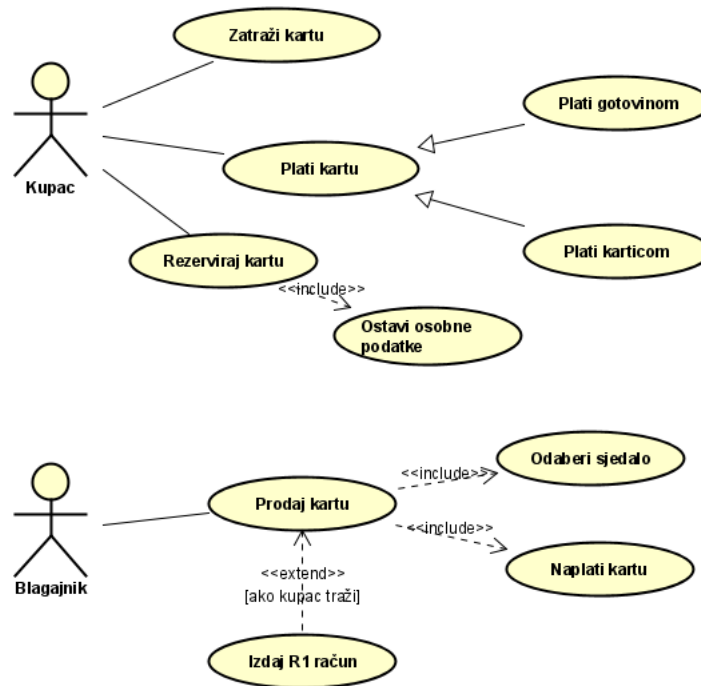
Tablica 1. Prikaz veza u dijagramu slučajeve korištenja

Izvor: vlastita izrada prema Seidl, (2012).

Asocijacijom se povezuju sudionici sa slučajevima korištenja. Npr. Kupac sudjeluje u odabiru proizvoda.

Generalizacijom se mogu označiti sudionici koji su vrsta drugog sudionika ili slučaj korištenja, koji je vrsta nekog drugog slučaja korištenja. Npr. sudionik društvo s ograničenom odgovornošću i dioničko društvo su vrste trgovačkog društva.

Uključivanje (engl. include) međusobno povezuje dva slučaja korištenja, odnosno prvi slučaj uključuje drugi slučaj korištenja. Proširenje (engl. extend) također međusobno povezuje dva slučaja, ali za razliku od uključivanja, jedan ne mora nužno uključivati drugi već proširuje funkcionalnost drugog i to samo ako je zadovoljen neki uvjet.



Slika 8. Primjer dijagrama slučajeva korištenja

Izvor: Vlastita izrada prema primjeru zadatka Šebalj (2020/2021).

Slika 8. prikazuje primjer korištenja dijagrama slučajeva korištenja na primjeru kupnje karte za kino. Kupac ima mogućnost traženja karte, njezinu kupnju te rezervaciju. Kartu može platiti gotovinom ili karticom, a kod rezerviranja karte je potrebno ostaviti osobne podatke. Blagajnik prodaje kartu, nudi kupcu mogućnost odabira slobodnog sjedala te mogućnost naplate karte. Kod prodaje karata, blagajnik može izdati R1 račun, ako to kupac zatraži.

**Dijagram klasa** potječe iz konceptualnog modeliranja podataka i objektno-orijentiranog razvoja sustava. Navedeni koncepti se koriste kako bi se specificirale struktura podataka i strukture sustava. Odnosno, dijagram klasa prikazuje sustav uz korištenje klasa (razreda) i relacija (veza) među klasama. Dijagram klasa je ujedno i najčešće korišteni dijagram. Za razliku od dijagrama slučajeva korištenja koji se koriste u ranim fazama razvoja, dijagram klasa se može koristiti u različitim fazama razvoja softvera, te se razina detalja ili apstrakcije razlikuje s obzirom na fazu. Što je kasnija faza razvoja, to će dijagram prikazivati više detalja. Definicija klase predstavlja osnovni element dijagrama klasa te ona opisuje skupine objekata sa sličnim svojstvima kao klasa. Objekt predstavlja instancu klase. Tako je, npr. Trgovačko društvo klasa, a Plodine, Konzum ili Infimum d.o.o. objekti te klase. Svaki objekt ima svoje atribute, odnosno svojstva i izvršavaju određene radnje (operacije/metode).

Tijekom kreiranja klase potrebno je odrediti naziv klase, te atribute i operacije koji su opcionalni. Naziv klase započinje velikim početnim slovom i u jednini. Atributi pružaju informacije o objektu, odnosno pružaju uvid u svojstvo klase, gdje prva riječ započinje malim, a druga velikim slovom.

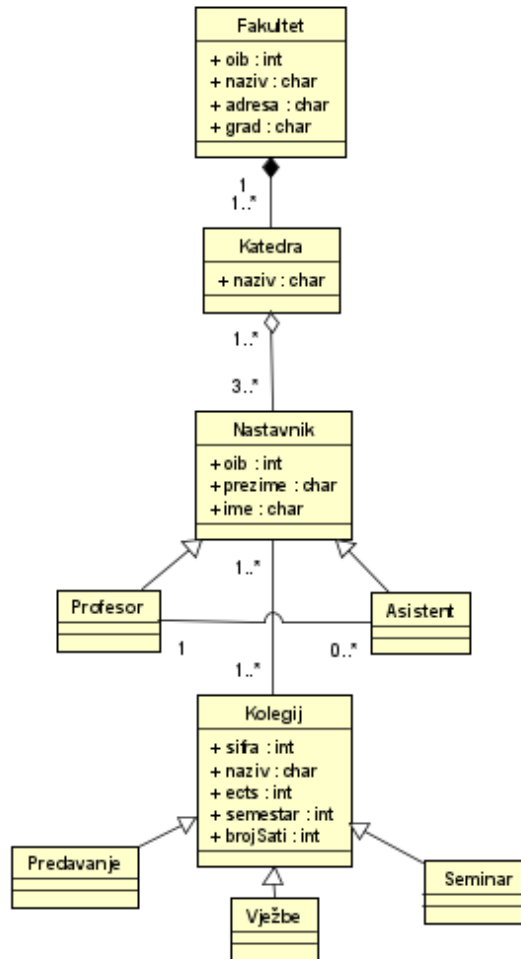
Atributi imaju četiri razine vidljivosti koje određuju tko sve ima mogućnost pristupiti atributima i operacijama klase. Njih je moguće vidjeti iz priložene tablice ispod.

VIDLJIVOST	SIMBOL	OPIS
Javno (engl. public)	+	Pristup ima objekt bilo koje klase
Privatno (engl. private)	-	Pristup ima samo promatrani objekt
Zaštićeno (engl. protected)	#	Pristup imaju objekti iste klase i njezine podklase
Paket (engl. package)	~	Pristup imaju objekti čije se klase nalaze u istom paketu

Tablica 2. Razine vidljivosti u dijagramu klasa

Izvor: vlastiti prijevod i izrada prema Seidl (2021, str. 59)

Vrste atributa moguće je podijeliti na dva tipa: UML tip (boolean, integer, string, unlimitedInteger) i java tipove (byte, char, double, float, int). Operacije su sve one stvari koje objekt ili klasa mogu raditi, odnosno izvršavati. Kao i kod atributa, operacije imaju razine vidljivosti te ulazne i izlazne parametre, koji su svojstveni samo za operacije. Parametri predstavljaju željene informacije koje su potrebne operacijama kako bi se izvršili zadaci. Odnose među klasama moguće je povezati vezama koje se međusobno razlikuju s obzirom na jačinu.



Slika 9. Primjer dijagrama klasa

Izvor: vlastita izrada prema primjeru od Šebalj (2020/2021)

Slika 9. prikazuje primjer korištenja dijagrama klasa na primjeru fakulteta.

#### 4.3.3. Dinamički modeli i primjeri upotrebe

Dinamički model obilježavaju slijedeći dijagrami:






- Sekvencijski dijagram (engl. Sequence Diagram)
- Dijagram kolaboracije (engl. Collaboration Use Diagram )
- Dijagram stanja (engl. State Diagram)
- Dijagram aktivnosti (engl. Activity Diagram).

„Dinamički modeli opisuju ponašanja elemenata i podsustava , promjene njihova stanja i međusobnu uvjetovanost“. (Mesarić, Šebalj, 2020)

U nastavku će se detaljnije opisati dvije vrste dinamičkih modela: sekvencijski dijagram i dijagram aktivnosti..

**Sekvencijski dijagram** odgovara na pitanje kako, odnosno opisuje kako će sustav napraviti neke zadatke. Sekvencijski dijagrami modeliraju važna vremenska izvođenja između dijelova koji čine sustav i dijelova koji čine logički pregled sustava, odnosno vremenskim redosljednom kako se odvija interakcija između određenih dijelova sustava, te kronološki pokazuje redosljed izvođenja operacija i događaja. U sekvencijskom dijagramu definiraju se sudionici (objekti) te poruke koje se odvijaju među njima.

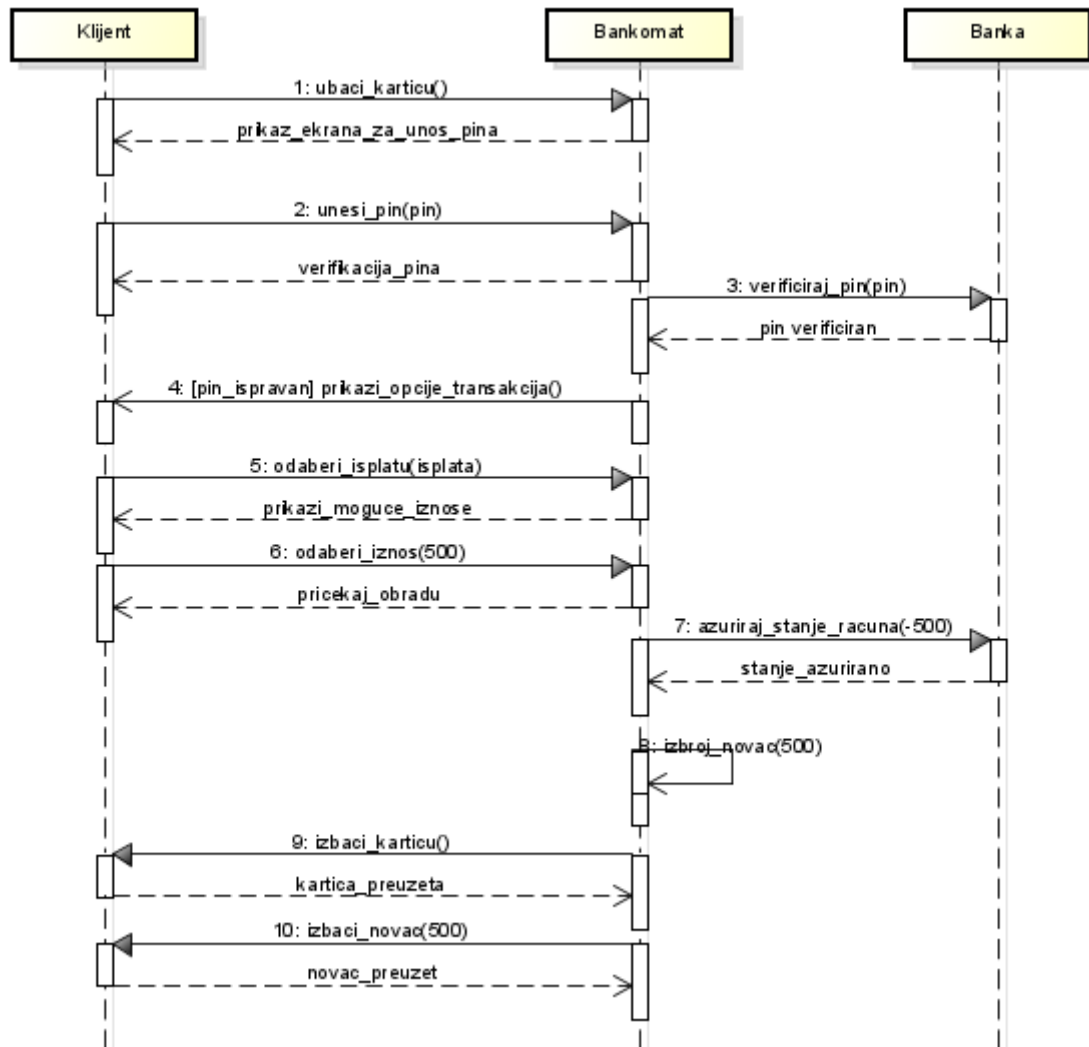
Sudionici (objekti) u sekvencijskom dijagramu mogu biti ljudi, kao npr. studenti ili profesori te mogu biti nežive stvari poput softvera, aplikacija i slično. (Seidl, 2012:107, prevela autorica).

PORUKA	IZGLED	OPIS
<b>Sinkrona poruka</b> (engl. Synchronous message)		Pošiljalac šalje poruku primatelju te pošiljalac ne očekuje odgovor primatelja.
<b>Asinkrona poruka</b> (engl. Asynchronous message)		Pošiljalac šalje poruku primatelju i čeka odgovor (povratnu vezu). Tek kad primatelj pošalje odgovor pošiljalcu se sve aktivnosti nastavljaju.
<b>Povratna poruka</b> (engl. Response message)		
<b>Kreiranje sudionika</b>		Sudionici (objekti) ne moraju biti živa bića, mogu biti neka aplikacija. Njih se može kreirati ili obrisati ovisno o porukama.
<b>Brisanje sudionika</b>		

Tablica 3. Vrste veza u sekvencijskom dijagramu

Izvor: vlastita izrada prema Šebalj (2020).

Između sudionika obavlja se određena komunikacija, gdje si sudionici međusobno šalju poruke koje su prikazane strelicama s različitim izgledom i namjenom, kako je vidljivo u tablici iznad.



Slika 10. Primjer sekvencijskog dijagrama

Izvor: Vlastita izrada prema zadatku od Šealj (2020/2021).

Slika 10. prikazuje korištenje sekvencijskog dijagrama. Dijagram prikazuje situaciju gdje klijent banke dolazi podići određeni iznos novca s bankomata.

**Dijagram aktivnosti** omogućuju modeliranje poslovnih procesa u informacijskom sustavu. Dijagram aktivnosti određuje protok kontrole i protok podataka među koracima, odnosno radnjama koje su potrebne za provođenje neke aktivnosti. Jedna od značajki dijagrama aktivnosti jest što podržava modeliranje objektno orijentiranih sustava i neobjektno

orijentiranih sustava. Dijagram aktivnosti, također, omogućuje definiranje aktivnosti neovisno o objektima. Ujedno je i jedini dijagram iz procesnog pregleda.

Kreiranje dijagrama aktivnosti započinje stavljanjem znaka početnog stanja (engl. *initial node*), koji je prikazan kao crno ispunjeni krug. Nema dolaznih tokova u početno stanje, ali mora imati odlazni tok (Šebalj, 2020/2021).

Završno stanje (engl. *final-activity node*) prikazuje završetak procesa (aktivnosti) te za razliku od početnog stanja, gdje nema dolaznih tokova, završno stanje ima samo dolazne tokove, ali ne i odlazne. Aktivnosti, odnosno zadaci predstavljaju pravokutnik zaobljenih rubova. Prijelaz aktivnosti je prikazan običnom strjelicom, te one povezuju jednu aktivnost s drugom i tako slijedno.

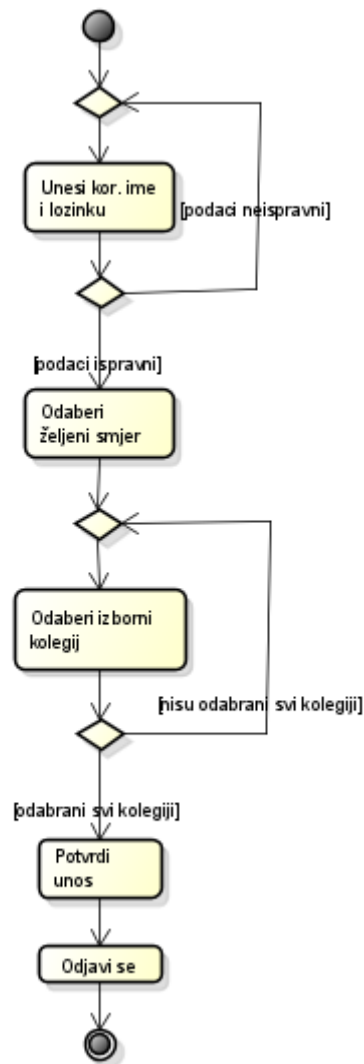
Čvor račvanja (engl. *parallelization node*) prikazan je kao crna okomita prepreka s jednim ulaznim tokom koji se šire u više paralelnih tokova. Čvor spajanja (engl. *synchronization node*) također prikazuje crnu okomitu prepreku, ali njezin je zadatak spojiti skup više paralelnih tokova u jedan zajednički tok.

Također, postoje i odluke (engl. *decision*), čiji izgled podsjeća na izgled romba s jednim ulaznim tokom i više izlaznih tokova. Odluke se koriste kada se želi izvršiti drugačiji redoslijed akcija, ovisno o stanju. Svi tokovi se na kraju spajaju u čvor spajanja (engl. *merge node*), koji izgleda isto kao i odluka, ali s više ulaznih tokova i jednim izlaznim tokom.

„Signali pokazuju da na neku aktivnost djeluju određeni događaji iz vanjskog procesa. Šaljući signali su signali koji su poslani vanjskom sudioniku, dok su primajući signali zaprimljeni od vanjskog sudionika.“ (Šebalj, 2020/2021)

Pojedine akcije mogu povezivati i druge aktivnosti, ako se želi napraviti još jedna aktivnost, koja neće biti prikazana u cjelokupnom dijagramu, ali će skriveno obavljati svoju funkciju i koristiti će se pozivanje drugih aktivnosti. Takva aktivnost izgleda kao normalna aktivnost, osim što u donjem desnom kutu ima znak koji podsjeća na vilicu. Klikom na taj zadatak, otvorit će se prozorčić za kreiranje novog, „pomoćnog“ dijagrama.





Slika 11. Primjer dijagrama aktivnosti

Izvor: vlastita izrada prema primjeru od Šebalj (2020/2021).

Slika 11. prikazuje korištenje dijagrama aktivnosti. Dijagram prikazuje primjer prijave u sustav ISVU radi odabira izbornog predmeta. Kreiranje dijagrama započinje početnim stanjem. Korisnik mora unijeti svoje korisničke podatke i lozinku, a ako su oni neispravni mora ponoviti postupak. Ako su mu podaci ispravni, prelazi se na sljedeći korak *Odaberi željeni smjer* te korisnik mora odabrati obvezne kolegije i izborni kolegij. Ukoliko osoba nije označila sve potrebne kolegije, neće moći potvrditi unos, već mora označiti sve kolegije. Nakon što je korisnik označio sve potrebne kolegije, uključujući izborni kolegije, korisnik može potvrditi unos i prijeći na posljednji korak *Odjavi se*. Kada je posljednji zadatak izvršen, stavlja se završno stanje.

## 5. Rasprava

Svaki model ima svoje kako koristi, tako i nedostatke koje se razlikuju od model do modela. Ne mogu sve organizacije koristiti iste modele. Tijekom razvoja nastajali su novi modeli poslovanja te su nestajali oni koji su pokazani da su neuspješni. Organizacija tijekom razvoja može promijeniti nekoliko modela, dok ne dođe do onog koji najviše odgovara njihovim zahtjevima u rješavanju problema i donošenja odluka u postizanju željenog cilja.

Dijagram slučajeva korištenja je jednostavan model, koji korisniku prikazuje jasan prikaz procesa i radnji koje bi sustav trebao izvršavati. Primarna uloga dijagrama slučajeva korištenja jest omogućiti razumijevanje zahtjeva, koji su postavljeni pred razvoj informacijskog sustava. Međutim, dijagram korištenja nije pogodan za detaljniju razradu zadataka i određenih pojedinosti svake stavke u procesu razrade koje bi sustav trebao izvršavati.

Dijagram klasa pogodan je za stavljanje detaljnih opisa zadataka koji ujedno koriste korisniku i analitičarima. Prikazuje detaljnu strukturu svakog pojedinog subjekta u dijagramu, njihov opis zadataka koji će izvršavati, te vrste svojstava stavki u klasi.

Snaga odnosa među klasama je bazirana na tome koliko klase koje su međusobno povezane ovise jedna o drugoj. Za klase koje su međusobno jako ovisne jedna o drugoj kaže se da su usko povezane, što znači da će promjene koje se dogode u jednoj klasi najvjerojatnije utjecati i na promjene u drugoj klasi. Uska povezanost je često loša stvar, ali ne uvijek. Što je veza između klasa jača, potrebno je paziti zbog promjena u klasama koje mogu nastati, ako se i samo jedna stavka promijeni. (Hamilton, 2021:109, prevela autorica).

Sekvencijski dijagram detaljno pokazuje korake koji korisnik prolazi kako bi izvršio željenu aktivnost koju očekuje od samog sustava. Ono detaljno pokazuje sve poruke koje šalju među objektima tijekom nekog vremenskog perioda. Sekvencijski dijagram, međutim, ne može prikazati više radnji u isto vrijeme, već prikazuje jednu aktivnost između najčešće dva objekta i to u određenom vremenskom periodu. Sekvencijski dijagram ima problem prikazivanja „if-else-than“ ili „for-next“ situacija. Zato se ono ne bi trebalo koristiti u opisivanju tih situacija. S obzirom da sekvencijski dijagram može prikazati jednu određenu aktivnost, potreban je veliki broj sekvencijskih dijagrama kako bi se prikazali svi procesi u sustavu. Zbog tog razloga, mnogi razvojni timovi odustaju od daljnjeg prikazivanja.

Jezici UML modeliranja koji se koriste u dijagramu aktivnosti čine ga jednim od najlakših dijagrama za korištenje i razumijevanje, kako za analitičare tako i krajnje korisnike, te se smatraju ključnima alatom kojim se analitičari vole koristiti. Zbog svoje jednostavnosti i razumljivosti kreiranja, dijagram aktivnosti se čini kao idealan dijagram u procesu razvoja cjelokupnog informacijskog sustava. Zbog svoje jednostavnosti, razvojni tim, koji kreiraju modele informacijskog sustava mogu napraviti cjelokupni dijagram upravo zato što je jednostavan, no više jednostavnih pojedinačnih modela stavljeni zajedno mogu načiniti jedan veliki kompleksan dijagram i model.

Dijagrami aktivnosti ne bi se smjeli koristiti umjesto dijagrama stanja ili sekvencijskog dijagrama, jer dijagram aktivnosti ne prikazuje detalje kako se objekti ponašaju, niti kako su oni povezani, odnosno kako surađuju. (ZeroDue Design, prevela autorica).

## 6. Zaključak

Kreiranje idealnog informacijskog sustava uz pomoć modela i pravog dijagrama je postao ključni zadatak za svaku organizaciju. Uz sve veći broj podataka koji su postali dostupni na internetu i u okolini, organizacije moraju prilagođavati svoje poslovanje novim trendovima. Kako bi organizacija kreirala svoj sustav, koji će izvršavati zahtjeve koji su postavljeni pred njim, mora proučiti modele i njihove dijagrame kako bi pronašla onaj idealni koji će na razumljiv, ali jednostavan način prikazati upravo ono što bi budući informacijski sustav trebao raditi, odnosno koje zadatke izvršavati. Tijekom pisanja rada, pisalo se o vrstama modela i vrstama dijagrama koji imaju svoje nedostatke, ali i koristi ukoliko se pravilno naprave i u pravilnoj fazi mogu itekako olakšati poslovanje organizacije. Tijekom pisanja rada i detaljne analize modela i dijagrama, došlo se do zaključka od strane autorice kako su dijagram slučajeve korištenja i dijagram aktivnosti jedni od onih dijagrama koje bi organizacije mogle obratiti pozornosti i odučiti se upravo za njih kod kreiranja sustava. Međutim, svaka organizacija sama uz pomoć stručnih osoba mora pronaći svoj vlastiti model i dijagrame, koji će moći koristiti u fazama procesa razvoja informacijskog sustava. Svaki model uz dijagrame nosi određene nedostatke, ali uz pravilnu vježbu, usavršavanje te učenje o određenom dijagramu se može pronaći onaj idealan.

## Literatura

### Knjige:

1. Čerić, V., Varga, M. (ur.); Bosilj Vukšić, V., Bubaš, G., Budin, A., Budin, L., Čerić, V., Dalbelo Bašić, B., Đurek, M., Hunjak, T., Indihar Štemberger, M., Jaklič, J., Lovrek, I., Pejić Bach, M., Radman, G., Varga, M. (2004). *Informacijska tehnologija u poslovanju*. Element, Zagreb
2. Hamilton, K., Miles, R., (2006). *Learning UML 2.0*, O'Reilly, Dostupno na: <http://jti.polinema.ac.id/wp-content/uploads/2019/02/Buku-Learning-UML-2.0.pdf> , [pristupljeno: 21.6.2021.]
3. Seidl, M., Scholz, M., Huemer, C., Kappel, G., (2012). *UML @ Classroom. An Introduction to Object-Oriented Modeling*. NJ: Springer. Dostupno na: [https://petcomputacao.ufsc.br/wp-content/uploads/2020/06/2015\\_Book\\_UMLClassroom.pdf](https://petcomputacao.ufsc.br/wp-content/uploads/2020/06/2015_Book_UMLClassroom.pdf) [pristupljeno: 23.6.2021.]
4. Srića, V., Spremić, Ma. (2000). *Informacijskom tehnologijom do poslovnog uspjeha*. Zagreb, Sinergija
5. Varga, M. i Strugar, I. (ur.); Pejić Bach, M., Varga, M., Srića, V., Spremić, M., Bosilj Vukšić, V., Ćurko, K., Vlahović, N., Milanović Glavan, Lj., Strugar, I., Zoroja, J., Jaković, B. (2016). *Informacijski sustavi u poslovanju*. Sveučilište u Zagrebu, Ekonomski fakultet
6. Velimir, S. (1981). *Sistem Informacija Kompjuter – Primjena sistemskih mišljenja u ekonomiji*, Informator, Zagreb (str. 15-16)
7. Gupta, U. *Information Systems*, Prentice Hall, 2000., str. 17

### Internetski izvori:

1. Mesarić, Josip. (2015/2016). *Informacijski sustavi u poslovanju – Ciljevi, zadaci i izgradnja informacijskih sustava*, Ekonomski fakultet u Osijeku. Dostupno na: [http://www.efos.unios.hr/informatika/wp-content/uploads/sites/202/2013/04/P11\\_Info\\_sustavi.pdf](http://www.efos.unios.hr/informatika/wp-content/uploads/sites/202/2013/04/P11_Info_sustavi.pdf) [pristupljeno: 14.6.2021.]

- Mesarić, J., Šebalj, D., *Oblikovanje i implementacija informacijskih sustava*, Nastavni materijali na predmetu Oblikovanje i implementacija informacijskih sustava, Ekonomski fakultet u Osijeku, ak.god. 2020/2021, raspoloživo na: <http://www.efos.unios.hr/oblikovanje-implementacija-is/predavanja/?lang=hr> [pristupljeno: 14.6.2021.]
2. Šebalj, D. *Modeliranje sustava i UML*, Nastavni materijali na predmetu Oblikovanje i implementacija informacijskih sustava, Ekonomski fakultet u Osijeku, ak.god. 2020/2021, raspoloživo na: <http://www.efos.unios.hr/oblikovanje-implementacija-is/wp-content/uploads/sites/219/2013/04/OIIS-Poglavlje-1.pdf> [pristupljeno: 14.6.2021.]
  3. Šebalj, D., *Dijagram klasa*, Nastavni materijali na predmetu Oblikovanje i implementacija informacijskog sustava, Ekonomski fakultet u Osijeku, ak.god. 2020/2021., raspoloživo na : <http://www.efos.unios.hr/oblikovanje-implementacija-is/wp-content/uploads/sites/219/2013/04/OIIS-Poglavlje-3.pdf> [pristupljeno: 25.6.2021.]
  4. Šebalj, D., *Sekvencijski dijagram*, Nastavni materijali na predmetu Oblikovanje i implementacija informacijskog sustava, Ekonomski fakultet u Osijeku, ak.god. 2020/2021., raspoloživo na : <http://www.efos.unios.hr/oblikovanje-implementacija-is/wp-content/uploads/sites/219/2013/04/OIIS-Poglavlje-5.pdf> [pristupljeno: 4.7.2021.]
  5. Šebalj, D., *Dijagram aktivnosti*, Nastavni materijali na predmetu Oblikovanje i implementacija informacijskog sustava, Ekonomski fakultet u Osijeku, ak.god. 2020/2021, raspoloživo na: <http://www.efos.unios.hr/oblikovanje-implementacija-is/wp-content/uploads/sites/219/2013/04/OIIS-Poglavlje-6.pdf> [pristupljeno: 5.7.2021.]
  6. ProgrammerSought , *UML software engineering modeling; Advantages and disadvantages of sequence diagram interaction and common errors in modeling and their corrective methods*, dostupno na: <https://www.programmersought.com/article/4869113487/> [pristupljeno: 9.7.2021.]
  7. Zerodue Design, *Activity Diagrams – Advantages, disadvantages and application of use*, dostupno na: <https://www.zeroduedesign.co.uk/blog/feed/activity-diagrams-advantages-disadvantages-and-applications-of-use/> [pristupljeno: 9.7.2021.]

## Popis slika i tablica

Slika 1. Konceptualni prikaz sustava .....	6
Slika 2. Prikaz sheme poslovnog informacijskog sustava.....	8
Slika 3. Primjer vodopadnog (engl. waterfall) pristupa .....	10
Slika 4. Primjer modificiranog vodopadnog pristupa .....	11
Slika 5. Primjer V – modela .....	12
Slika 6. Prikaz kategorija dijagrama 2.5 UML verzije.....	15
Slika 7. Krunchtenov 4+1 model.....	16
Slika 8. Primjer dijagrama slučajeve korištenja .....	19
Slika 9. Primjer dijagrama klasa .....	21
Slika 10. Primjer sekvencijskog dijagrama .....	23
Slika 11. Primjer dijagrama aktivnosti .....	25
Tablica 1. Prikaz veze u dijagramu slučajeve korištenja .....	18
Tablica 2. Razine vidljivosti u dijagramu klasa .....	20
Tablica 3. Vrste veza u sekvencijskom dijagramu .....	22