



TEHNIČKO VELEUČILIŠTE U ZAGREBU
POLYTECHNICUM ZAGRABIENSE

Objektno orijentirani razvoj programa

ISVU: 130938/19881

Dr. sc. Danko Ivošević, dipl. ing.

Predavač

Akademska godina 2021./2022.
Ljetni semestar

OBJEKTNO ORIJENTIRANI RAZVOJ PROGRAMA

UML DIJAGRAMI

KOMPONENTI

Dijagram komponenti

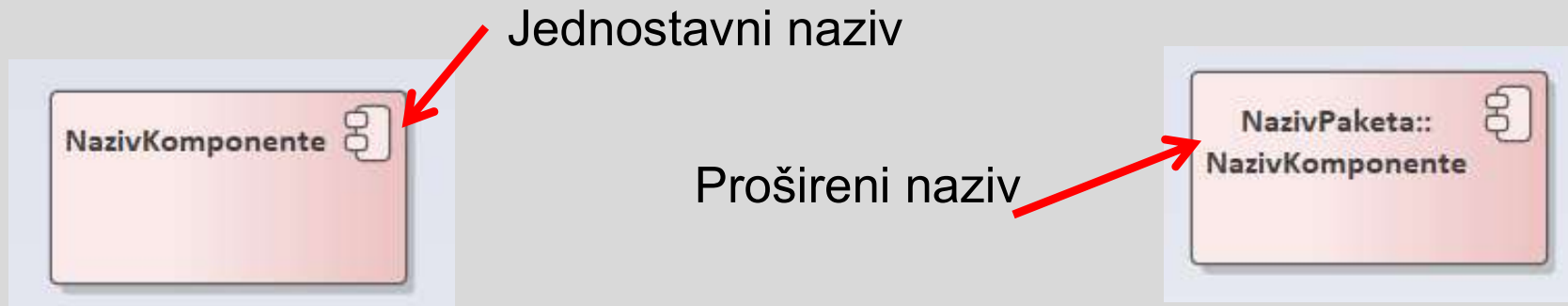
- Prikaz arhitekture programske potpore
 - gotovi dijelovi ili strukturne cjeline sustava i njihovi međusobni odnosi.
- **Strukturni UML dijagram, opisuje statične odnose:**
 - organizaciju i međuovisnost između implementacijskih komponenti programske potpore
- Zajedno s dijagramima razmještaja nazivaju se **fizički dijagrami:**
 - Pojedinci komponenti nalaze se u dijagramu razmještaja.

Komponente

- Komponenta \equiv zasebna cjelina programske potpore s vlastitim sučeljem
- Komponenta \equiv fizička i stvarna implementacija logičkih elemenata sustava
- \rightarrow Fizičke cjeline sustava
- \rightarrow Izvršne datoteke, programske knjižnice, tablice, datoteke i svi drugi dokumenti.
- Može sadržavati druge komponente \rightarrow prikaz unutarnje strukture

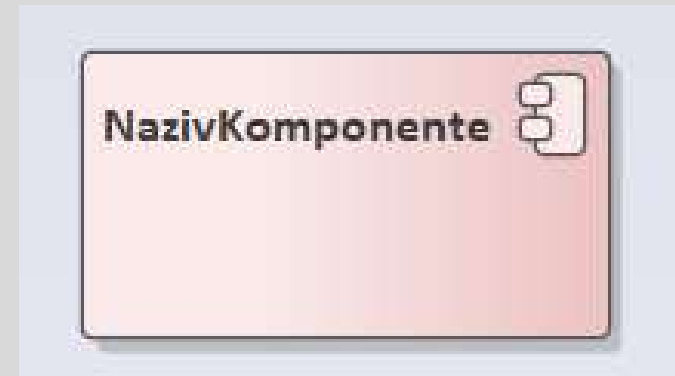
Svojstva komponenti

- Svaka komponenta ima jedinstven naziv ili ime koje je razlikuje od ostalih komponenti u dijagramu.
 - **Jednostavni naziv** (engl. *simple name*): niz znakova s određenim značenjem za korisnike dijagrama. U praksi uključuju i nastavak njihovih datoteka.
 - **Naziv puta** (engl. *path name*): naziv paketa kojemu komponenta pripada. Zapisuje se kao prefiks jednostavnom nazivu.

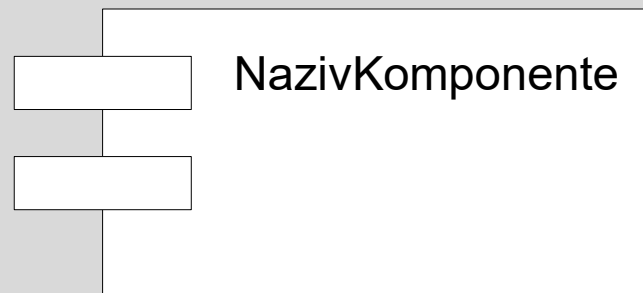


Notacija komponente

- Od UML standarda 2.0:

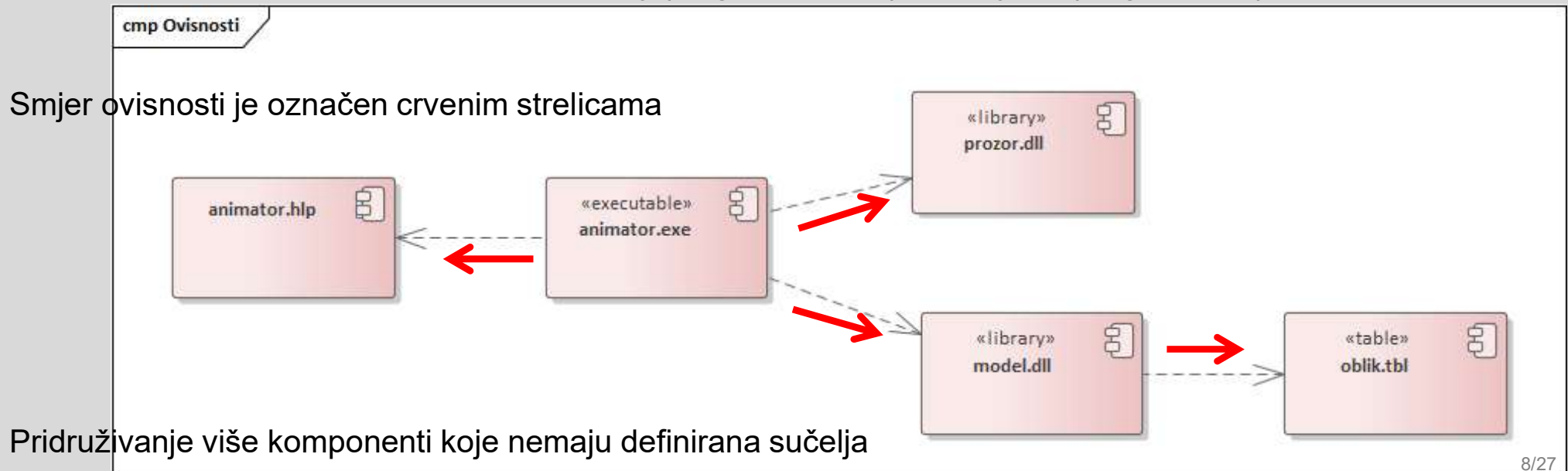


- U standardu 1.x:



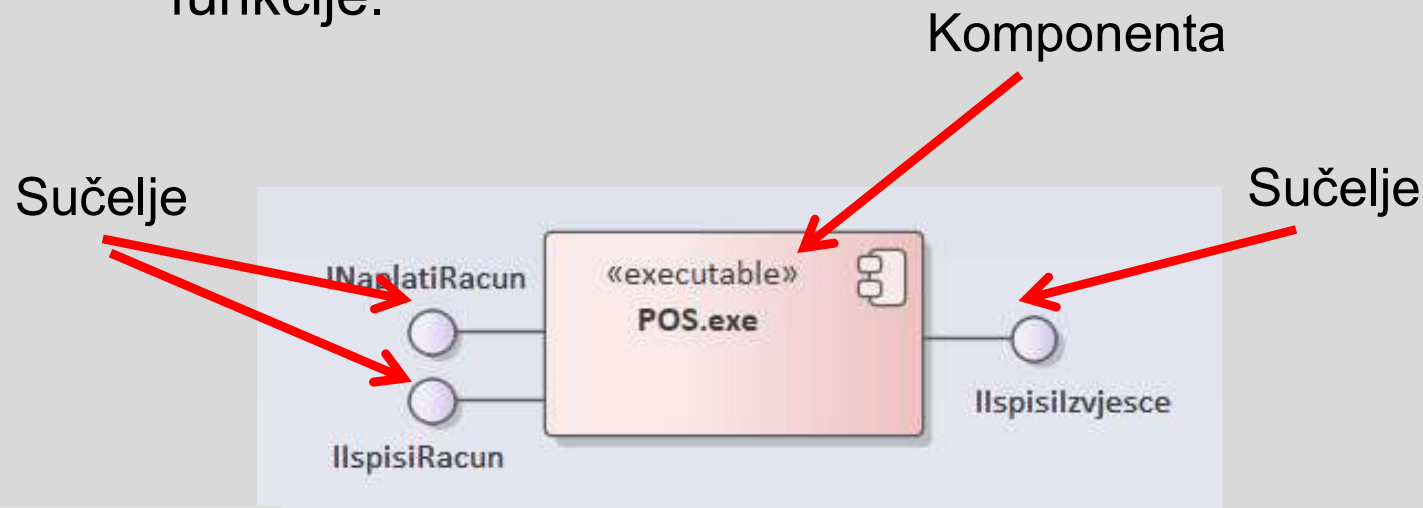
Pridruživanje komponenti

- **Pridruživanje** (engl. *association*) ili **veza** opisuje odnose između komponenti ili pojedinaca komponenti.
 - Komponente se pridružuju sučeljima ako su definirana.
 - Dijagram komponenti u pravilu koristi jednu vrstu veze: **ovisnost** (engl. *dependency*).
 - Ovisnost je uvijek jednosmjerna: “**B ovisi o A**” u smjeru strelice.
 - **A** se naziva isporučitelj (engl. *supplier*) i **B** klijent (engl. *client*).



Sučelja komponenti (1)

- Sučelje je kolekcija operacija koja određuju usluge komponente.
- Ponašanje komponente definirano je pomoću njezinih sučelja. Komponenta može biti zamijenjena drugom komponentom ako i samo ako su njihova sučelja identična. To je temeljena ideja ponovnog iskorištenja koda (engl. *software reuse*).
 - U UML dijagramima uporaba sučelja nije uvijek nužna. Komponente se mogu povezati neposredno. Ali zbog veće izražajnosti dijagrama potrebno je odrediti sučelja i njihove funkcije.

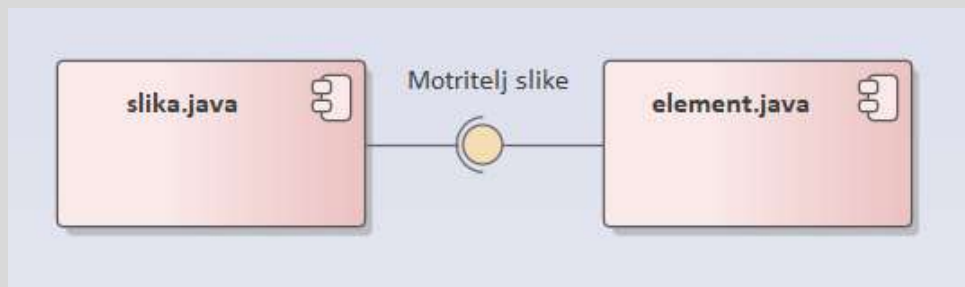


Sučelja komponenti (2)

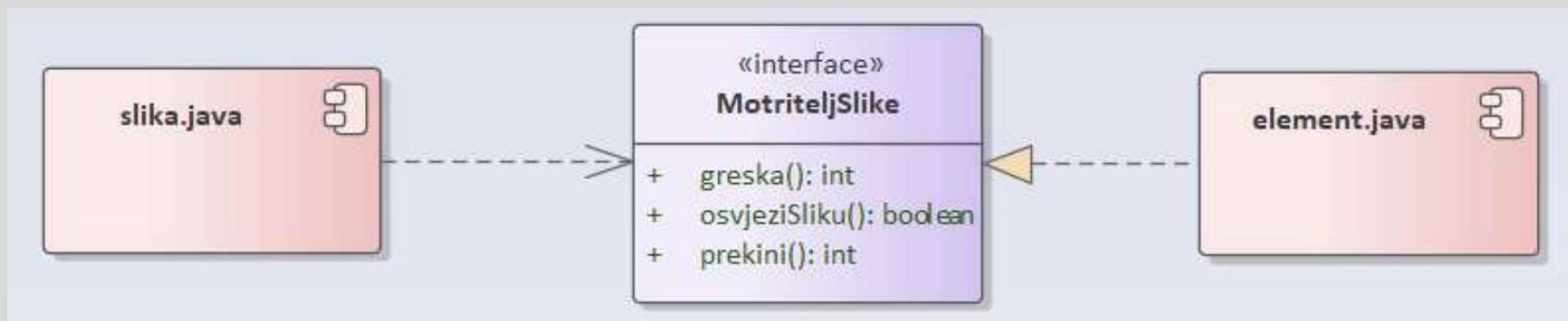
- Sučelja komponenti mogu biti prikazana na dva načina:
 - **Ikonizirani oblik** (engl. *iconic form*). Najčešći način prikaza sučelja. Komponenta koja realizira sučelje povezana je sa sučeljem s izostavljenom vezom realizacije (engl. *elided realization relationship*).
 - **Prošireni oblik** (engl. *expanded form*). Prikaz sučelja pomoću veze realizacije. Izgled ove veza identičan je vezi realizacije u dijagramima razreda. Operacije sučelja **moraju biti prikazani** u dijagramu.

Sučelja komponenti (2)

- Ikonizirani oblik

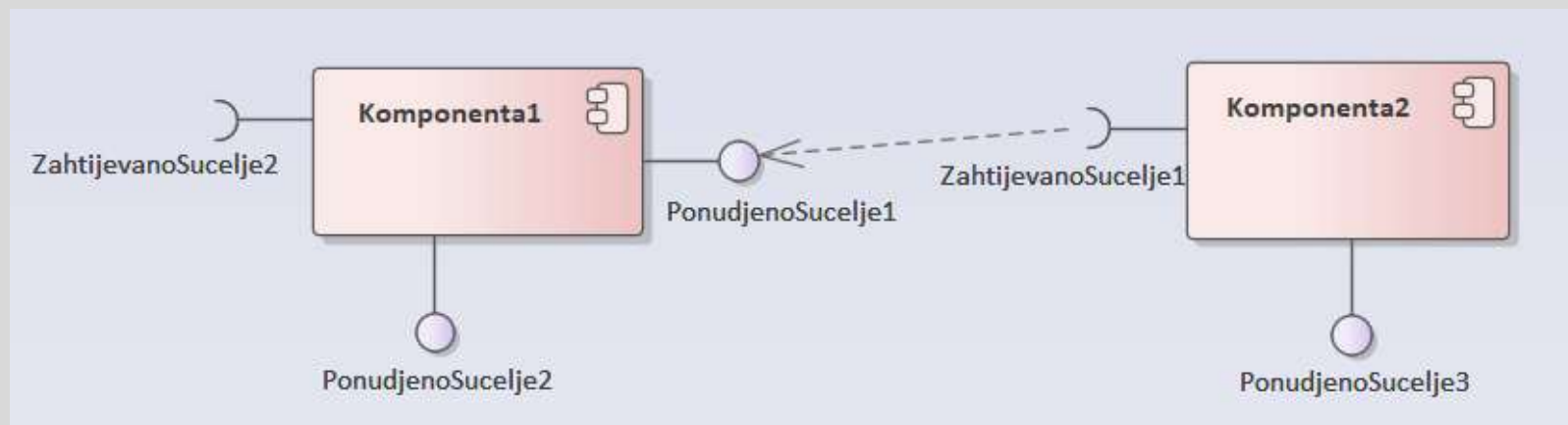


- Prošireni oblik



Sučelja komponenti (3)

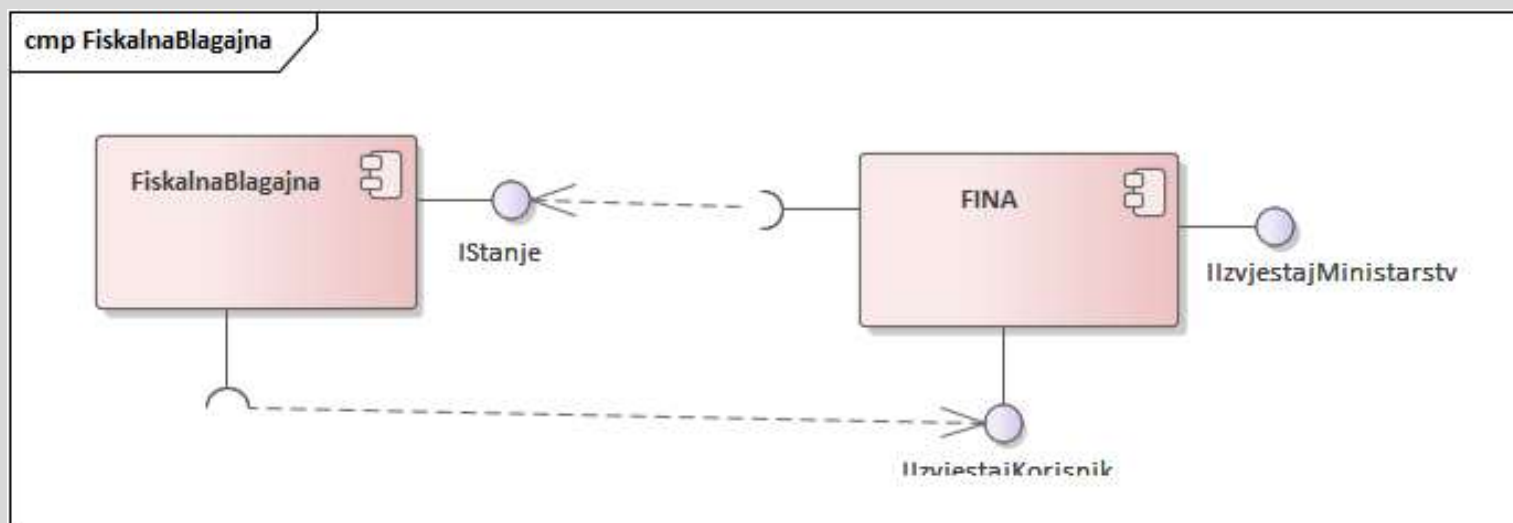
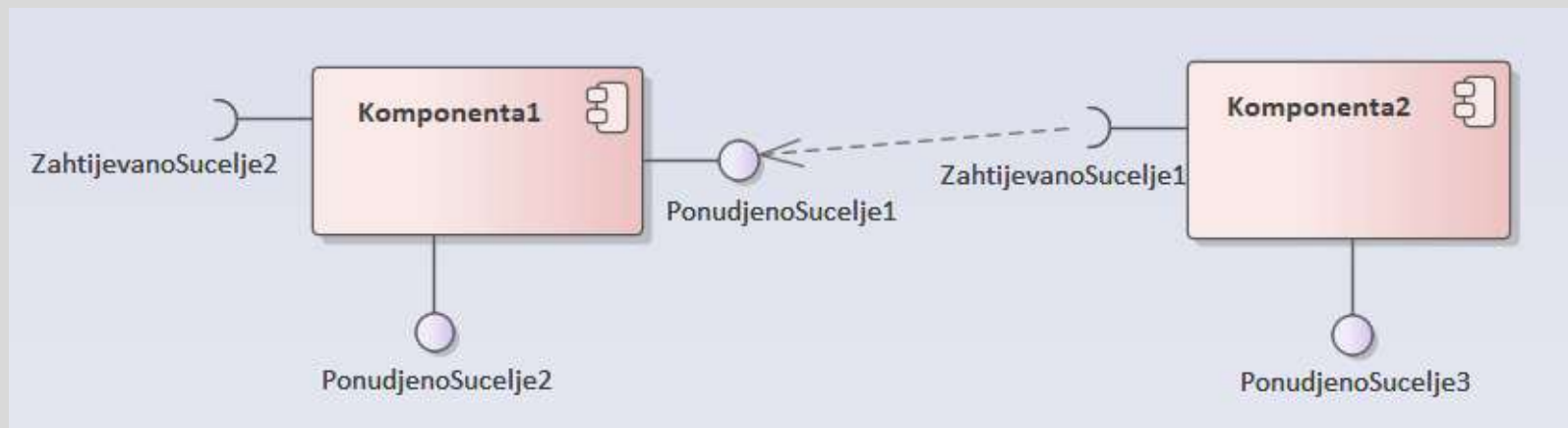
- **Ponuđeno sučelje** (engl. *export* ili *provided interface*) je sučelje koje neka komponenta realizira kao uslugu za druge komponente.
- **Zahtijevano sučelje** (engl. *import* ili *required interface*) je sučelje koje neka komponenta koristi.
 - Jedna komponenta može realizirati više ponuđenih sučelja i koristiti neograničeni broj zahtijevanih sučelja.



Višestruka ponuđena i zahtijevana sučelja komponenti

Sučelja komponenti (3)

- Primjeri – višestruka ponuđena i zahtijevana sučelja komponenti:



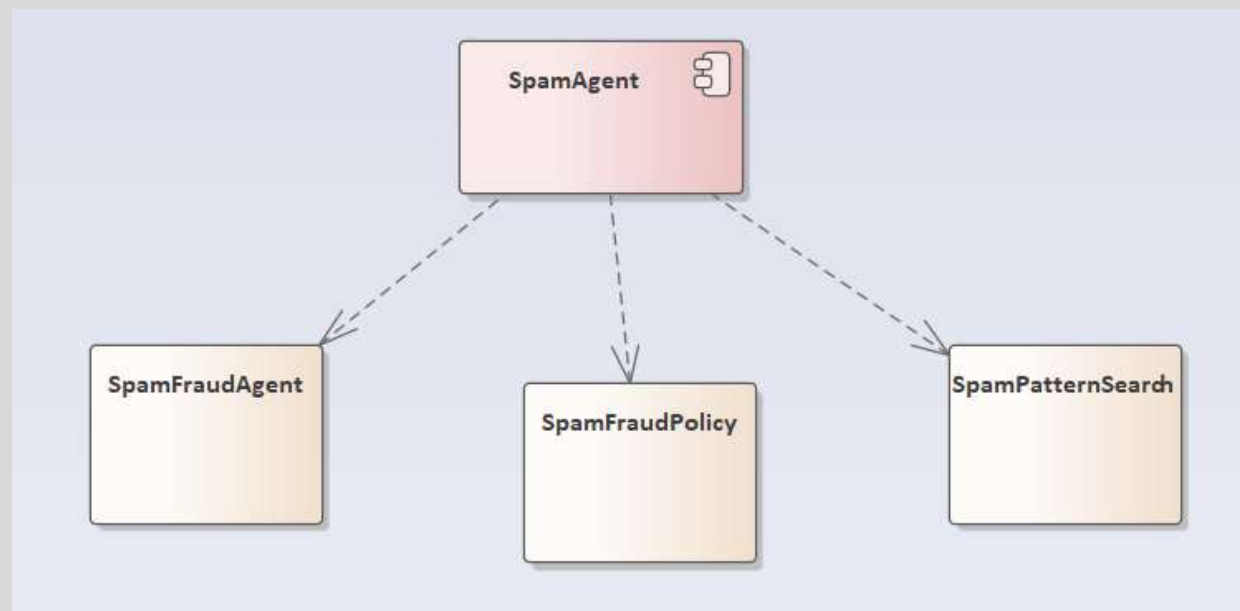
Razlike komponenti i razreda (1)

- Razlike komponenti i razreda (iz UML dijagrama razreda) su:
 - **Razina apstrakcije:** razredi predstavljaju logičku apstrakciju sustava, dok su komponente fizički (stvarni i opipljivi) artefakti.
 - **Pogled na sustav:** komponente i razredi pripadaju različitim pogledima na sustav s drugačijim razinama apstrakcije.
 - **Funkcionalnost:** razredi imaju vlastite attribute i operacije koje mogu neposredno izvršavati. Komponente imaju pristup samo onim operacijama koje se mogu dosegnuti kroz sučelja.

Odnos komponenti i razreda (2)

- U iznimnim slučajevima moguće je dijagramom komponenti prikazati razrede koji tvore komponentu.
 - Koristi se veza ovisnosti.

Primjer: DLL-datoteka spamagent.dll sastoji se od tri razreda: SpamAgent, SpamPolicy i SpamPatternSearch.



Vrste komponenti (1)

- UML definira tri vrste komponenti:
 1. **Komponente isporuke/razmještaja** (engl. *deployment components*)
 2. **Komponente radnog proizvoda** (engl. *work product component*)
 3. **Komponente izvođenja** (engl. *execution components*)
- Imaju isti UML simbol, ali razlikuju se po stereotipovima.

Vrste komponenti (2)

- **Komponente razmještaja/isporuke** (engl. *deployment components*) su one komponente koje su **nužne i dovoljne** za ispravan rad dizajniranog sustava.
- Primjeri:
 - Dinamički povezane biblioteke (engl. *dynamic linked libraries*, DLL) i izvršne datoteke (engl. *executables*, EXE).
 - Također: COM+, CORBA, Enterprise Java Bean, tablica baza podataka i druge izvršne datoteke.

Vrste komponenti (3)

- **Komponente radnog proizvoda** (engl. *work product component*) su one datoteke koje su **nastale tijekom razvoja sustava.**
- Primjeri:
 - Datoteke s izvornim kodom (engl. *source code*)
 - Datoteke s podacima (engl. *data files*)
- Komponente radnog proizvoda koriste se za realizaciju drugih komponenti.

Vrste komponenti (4)

- **Komponente izvođenja** (engl. *execution components*) nastaju kao **posljedica izvođenja i rada sustava**.
- Primjer:
 - COM+ objekti jer nastaju izvođenjem DLL datoteka.
 - Program Debug Database (PDB) datoteke nastale tijekom izrade .NET projekata.
 - Vlastite posredne (engl. *intermediate*) datoteke bilo kojeg formata.

Stereotipovi komponenti

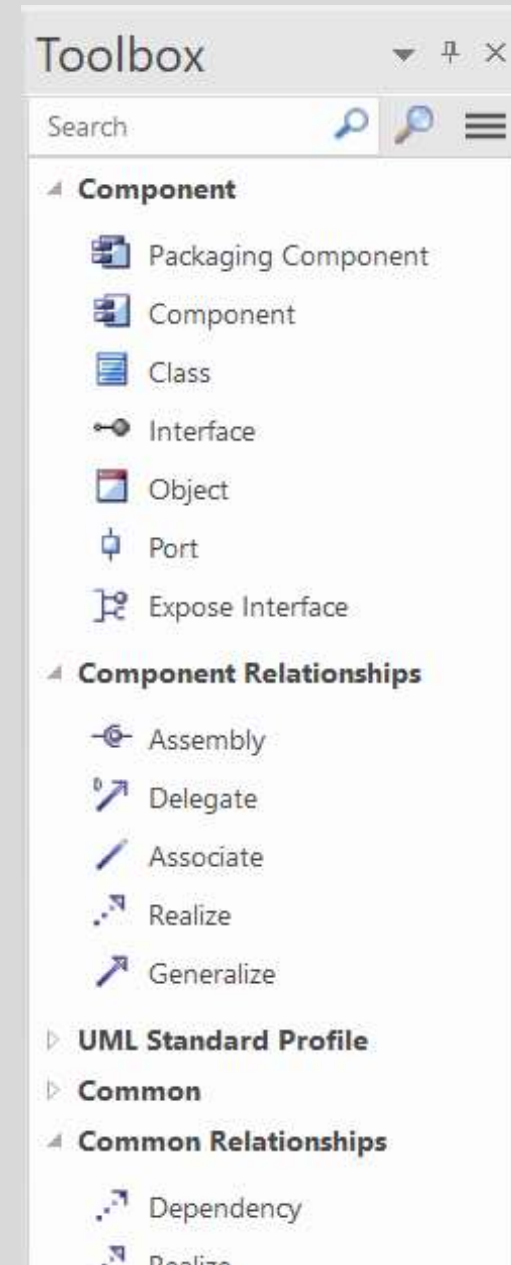
- Primjeri stereotipova za komponente:
 - «**executable**» - Komponenta može biti izvršena u čvoru
 - «**library**» - Komponenta je statička ili dinamička objekta biblioteka
 - «**table**» - Komponenta je tablica baze podataka
 - «**file**» - Komponenta je dokument s izvornim kodom ili podacima
 - «**document**» - Komponenta je dokument općeg značenja ili sadržaja
- UML ne definira specifične ikone za stereotipove, ali potiče se korištenje u praksi standardiziranih slikovnih simbola za izvršne datoteke, DLL datoteke, tablice baze podataka, itd.

Dijagram komponenti - Općenito

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

Dijagram komponenti u EA 15

- Komponenta
- Sučelje
- Sklapanje (engl. *assembly*)
- Realizacija
- Ovisnost



Paketi: Organizacija komponenti u dijagramu

- **Komponente se hijerarhijski organiziraju i grupiraju u pakete** slično kao što se organiziraju razredi.
- UML v1.x dopušta organizaciju komponenti **pomoću pridruživanja (veze) ovisnosti**.

Primjer: Modeliranje komponenti sustava

Nakon instalacije, Windows aplikacija za računalnu animaciju „Animator“ sastoji se od izvršne datoteke 'animator.exe' koja koristi DLL datoteke 'prozor.dll', 'model.dll', 'crtaj.dll' i 'prikazi.dll'.

Konfiguracija aplikacije nalazi se u formatiranoj datoteci 'animator.ini', a sustav pomoći u binarnoj datoteci 'animator.hlp'.

Izvršna datoteka 'animator.exe' je u inačici 3.1.4.

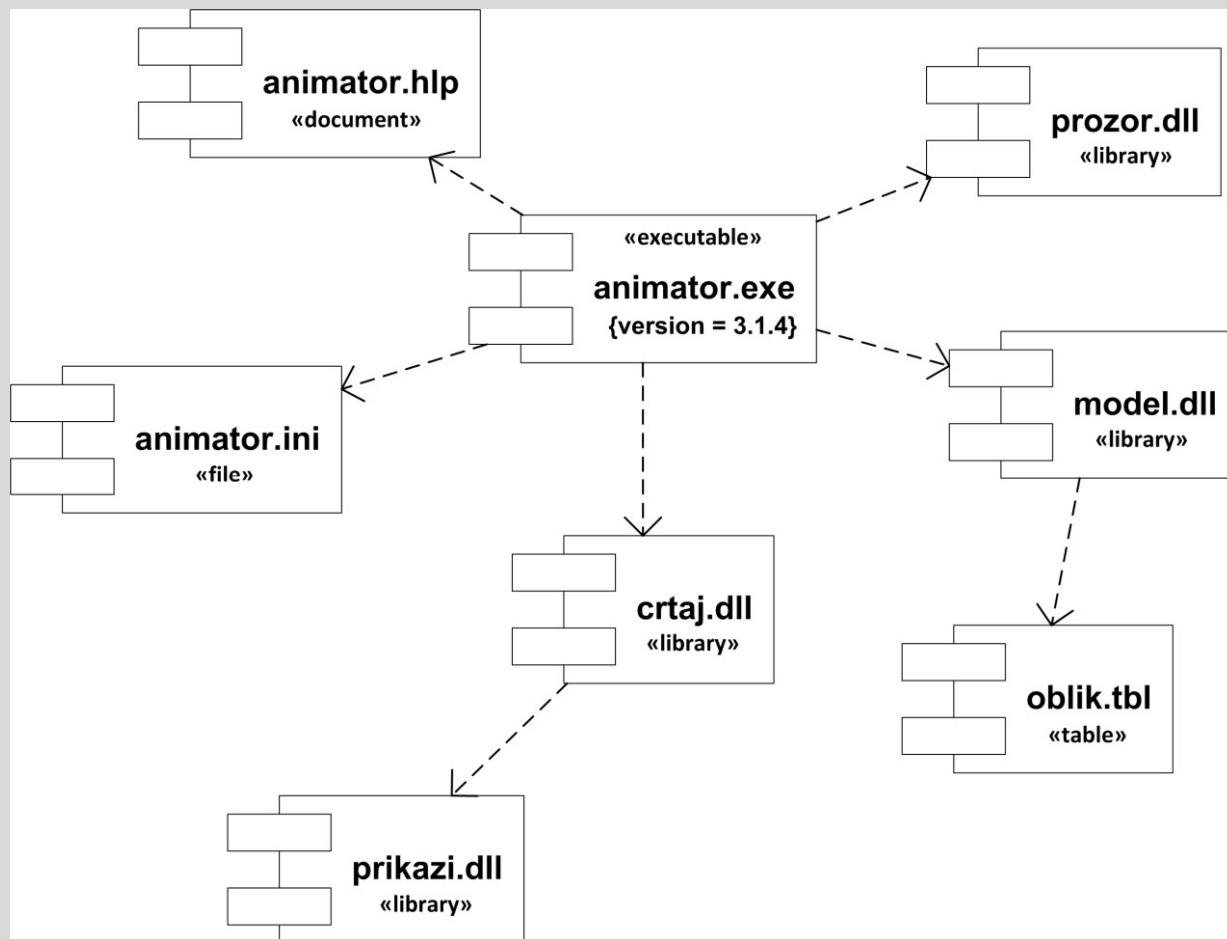
Biblioteka 'model.dll' pohranjuje podatke u tablicu baze podataka pod nazivom 'Oblik', koja se sastoji od datoteke 'oblik.tbl'.

Biblioteka 'crtaj.dll' koristi 'prikazi.dll'.

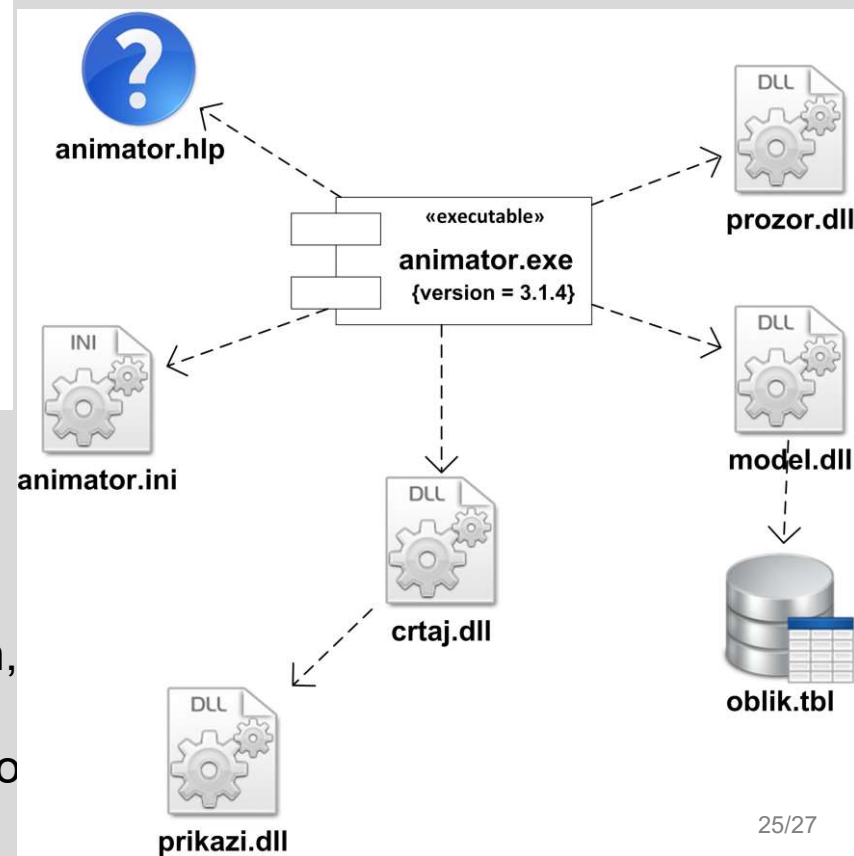
Prikažite navedene komponente u dijagramu.

Rješenja primjera

Koriste se stereotipovi komponenti i standardni simboli za prikaz datoteka. Jedina veza u dijagramu je ovisnost.



Zbog jasnoće dijagrama često se koriste normirane ikone umjesto simbola UML-komponenti. Ako je dijagram složen, s puno komponenti i veza, upotreba ikona pridonosi jednostavnijem i bržem razumijevanju dijagrama, pogotovo kod osoba koje nisu stručne u UML-u.



Laboratorijske vježbe

UML DIJAGRAMI KOMPONENTI U DOKUMENTU OSTVARENJA

Projektna dokumentacija

- moraju biti zastupljeni svi UML dijagrami o kojima se govorilo na predavanjima:
 - jedan ili više dijagrama obrazaca uporabe kojima se prikazuje cijeli sustav
 - najmanje dva dijagrama aktivnosti (ili jedan dijagram stanja) i koliko god je potrebno dijagrama slijeda za opis svih obrazaca uporabe
 - jedan ili više dijagrama razreda
 - jedan ili više dijagrama komponenti
 - dijagram razmještaja

REFERENCE I LITERATURA

- Sveučilišna zbirka zadataka iz UML-a - A. Jović, M. Horvat, I. Grudenić, “UML-dijagrami, zbirka primjera i riješenih zadataka”, 2014., *dostupno u Skriptarnici i knjižnici Fakulteta elektrotehnike i računarstva, te Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.*
- Predavanja ovog predmeta
- Allen Holub's UML Quick Reference:
<http://www.holub.com/goodies/uml>
- ArgoUML manual:
<http://argouml.tigris.org>
- Booch G., Jacobson I., Rumbaugh J. “UML Distilled”.
- Prezentacije predavanja i drugi dokumenti kolegija Oblikovanje programske potpore, dodiplomski studij, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu.