



УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

СТРУКТУРНА СИСТЕМСКА АНАЛИЗА

- Структурна системска анализа (ССА) је једна потпуна методологија за спецификацију информационог система.
- Она се на различите начине може повезати са методама других фаза у неку специфичну методологију целокупног развоја ИС.
- ССА је полазна основа за методу структурног пројектовања програма.

- Третира се као методолошки поступак декомпозиције неког система на подсистеме.
- Потпуна, тачна, формална и јасна спецификација ИС представља захтеве које будући систем треба да задовољи и предуслов за успешно даље пројектовање и имплементацију система.
- Спецификација је основа за "трансформационо" пројектовање и имплементацију, за аутоматизовано генерисање базе података и програма из ње, односно за коришћење CASE система.

- Оригинална ССА поседује веома једноставне, графичке и јасне концепте.
- Спецификација ИС приказује **ШТА** ИС треба да да, а **НЕ** и **КАКО** то треба да оствари.
- Одговор на питање "**како**" даје се за конкретно окружење, за дефинисану технологију и организацију у којој се систем имплементира.
- ССА не би требало да садржи технолошка и организациона ограничена решења, треба да се опише функционисање ИС у "идеалној технологији".

- Основни концепти за спецификацију ИС у ССА су:
 - функције, односно процеси обраде података,
 - токови података,
 - складишта података
 - и интерфејси.
- Међусобни однос основних концепата се приказује преко дијаграма тока података.

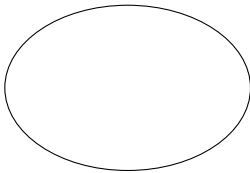
- ССА посматра информациони систем као функцију (процес) која, на бази улазних, генерише излазне податке.
- Улазни подаци се доводе у процес обраде, а излазни из њега одводе преко токова података.
- Ток података укључује:
 - папирне документе,
 - низ порука које човек уноси преко тастатуре терминала,
 - "пакет" информација добијен преко неке телекомуникационе линије или слично.

Основни концепти ССА

Складишта података, Интерфејси



- Извори излазних, односно понори улазних токова података унутар система називају се **складишта**.
- Складишта података се посматрају као "**токови у мировању**":
 - различите врсте евиденција,
 - архива,
 - картотека и
 - датотека
- Објекти који се налазе изван ИС а који са ИС комуницирају називају се **интерфејси**.
- И за складишта као и за токове од интереса су искључиво њихов садржај и структура.



Функција или процес обраде података



Интерфејс

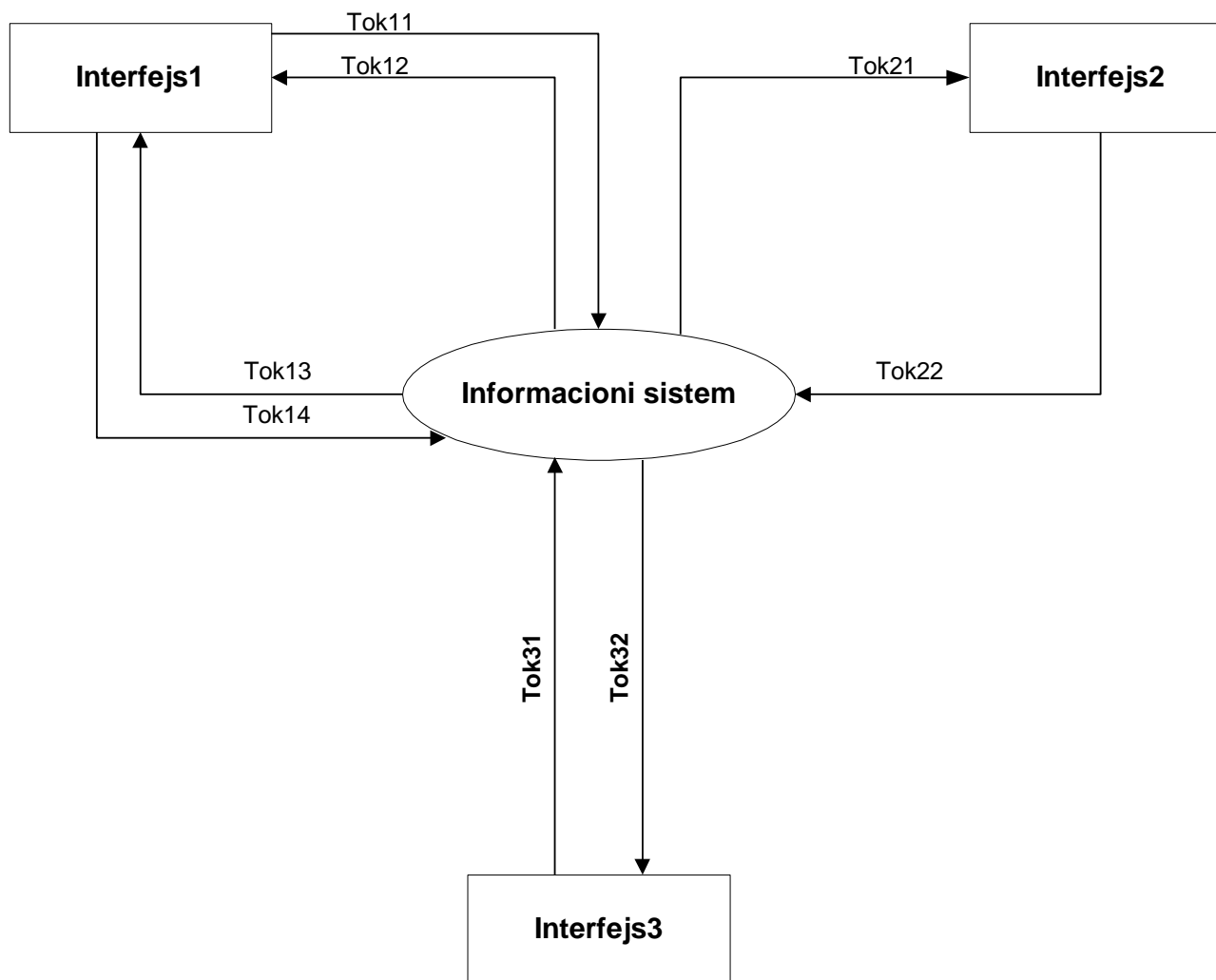


Ток података

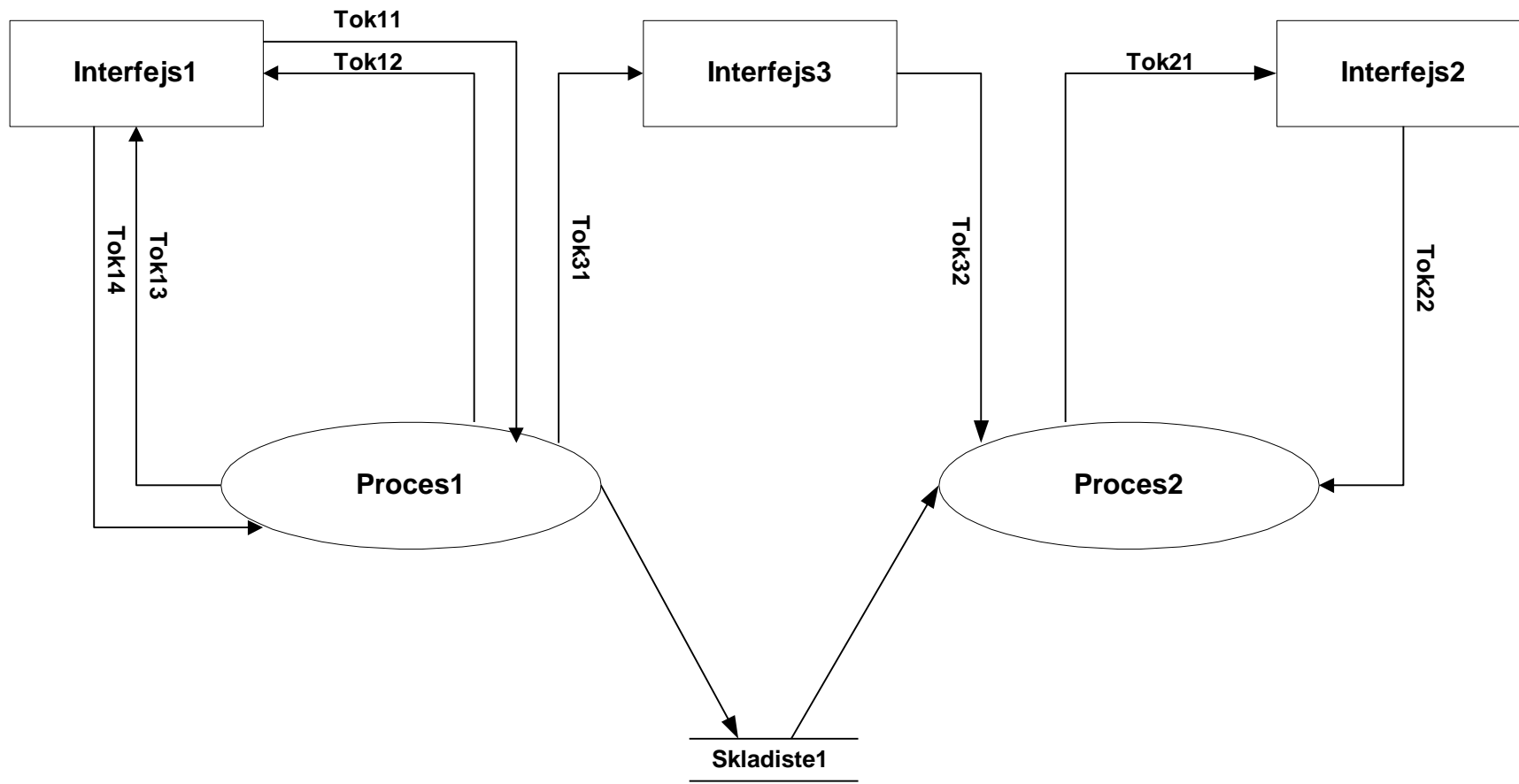


Складиште података

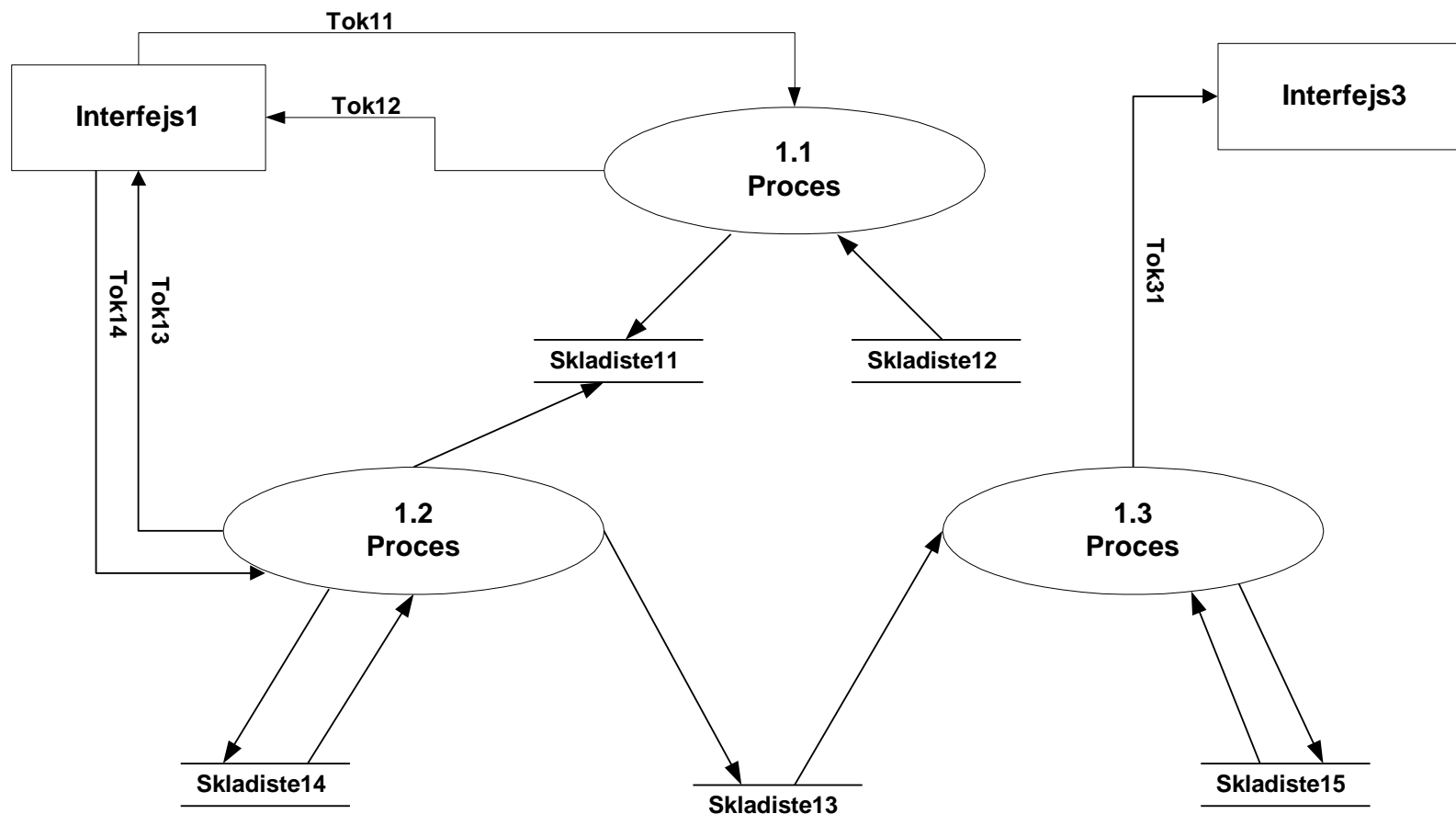
Пример дијаграма тока података



Пример дијаграма тока података



Пример дијаграма тока података



- Један ИС састоји из мноштва процеса, интерфејса, токова и складишта података.
- Детаљан и јасан опис система захтева опис на "различитим нивоима апстракције"
- Хијерархијским описом се на вишим нивоима систем описује општије (глобални процеси), а на нижим се постепено процеси декомпонују и потпуно и детаљно разрађују (до примитивних процеса).

- Дијаграм тока података на врху овакве хијерархије назива се **ДИЈАГРАМ КОНТЕКСТА**.
- Процеси на најнижем нивоу (процеси који се даље не декомпонују) називају се **ПРИМИТИВНИ ПРОЦЕСИ**.
- Једну потпуну спецификацију ИС чине:
 1. Хијерархијски организован скуп дијаграма тока података;
 2. Речник података који описује садржај и структуру свих токова и складишта података;
 3. Спецификација логике примитивних процеса;

ДИЈАГРАМИ ТОКА ПОДАТАКА

- Увод у ДТП
- Правила креирања ДТП-а
- Хијерархијска декомпозиција ДТП-а
- Пример ДТП

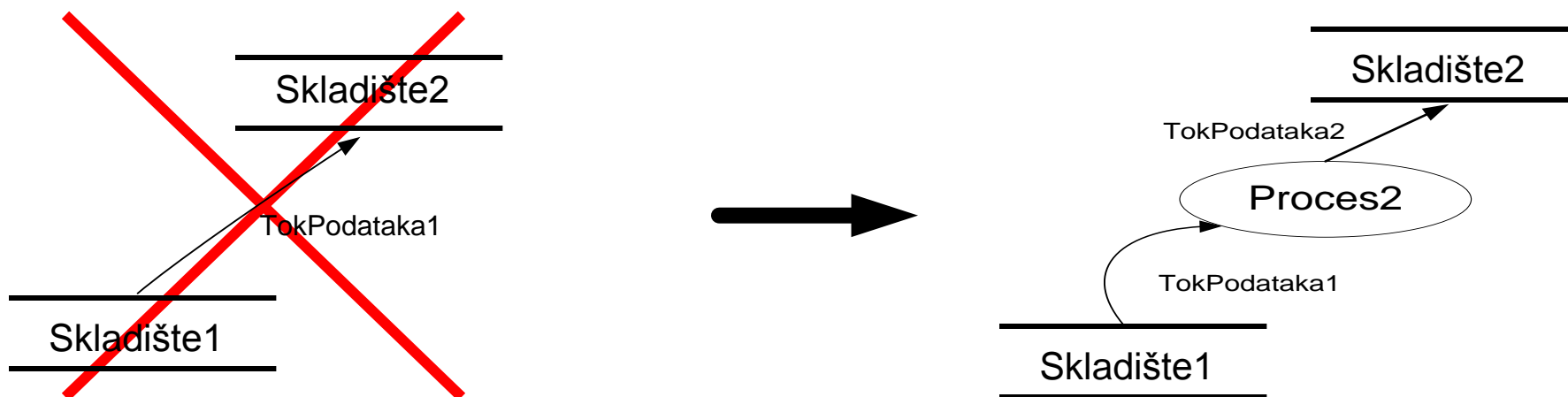
- Представља модел система који садржи четири основне компоненте:
 - **процесе обраде података** (активне компоненте система),
 - **објекте окружења** (интерфејсе) са којима систем комуницира,
 - **складишта података** које процеси користе и/или ажурирају и
 - **токове података** који повезују остале компоненте система у целину.
- Основне карактеристике ДТП-а су:
 - **јасна графичка спецификација**, погодна за комуникацију са корисником,
 - **јасан и детаљан опис система**, применом методе апстракције

- Токови података могу да буду :
 - различита документа,
 - формулари,
 - текстови,
 - књиге,
 - часописи и
 - слично.
- Ток података остварује везу између осталих компоненти система и на ДТП-у се представља именованом, оријентисаном линијом.

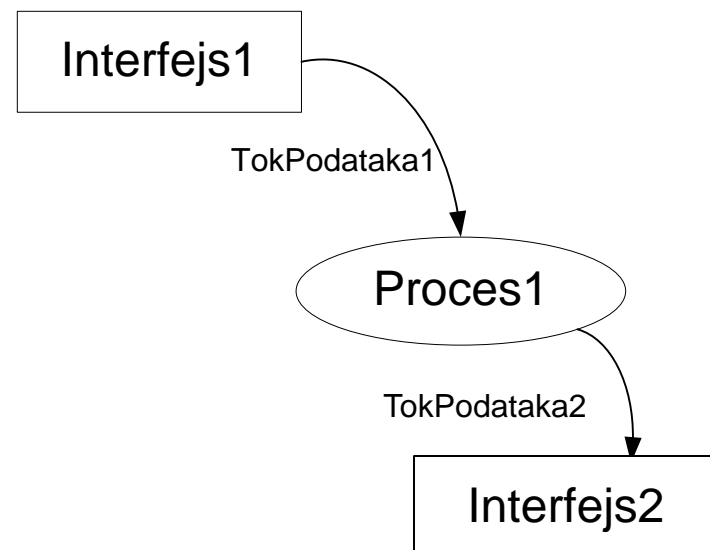
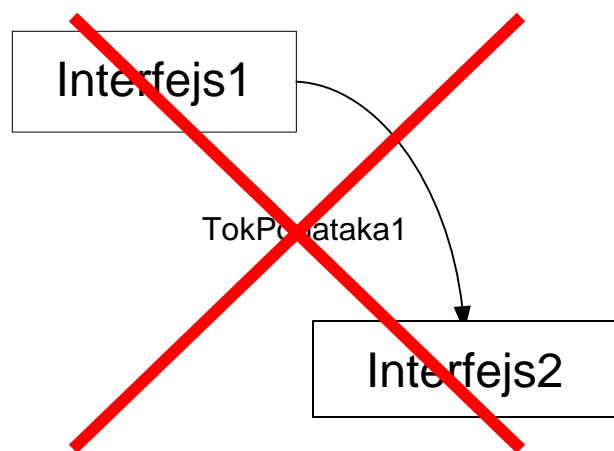
- Ток података мора да има извор и понор.
- Било која компонента ДТП може да буде улаз или излаз.
- Токовима се не могу непосредно повезати два складишта, два интерфејса, или складиште и интерфејс.

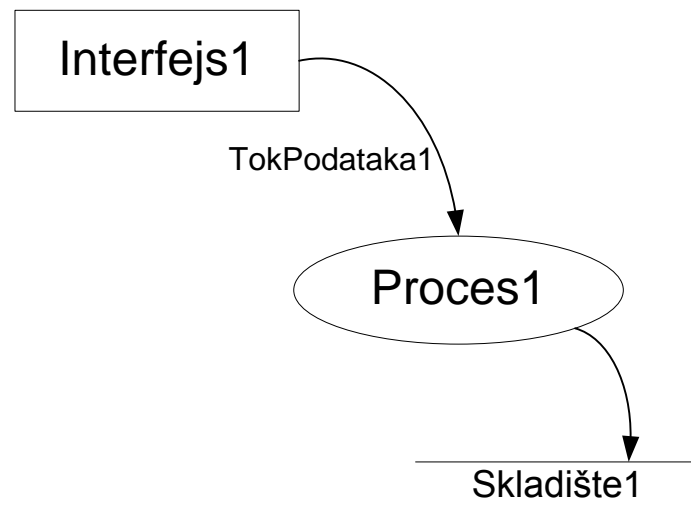
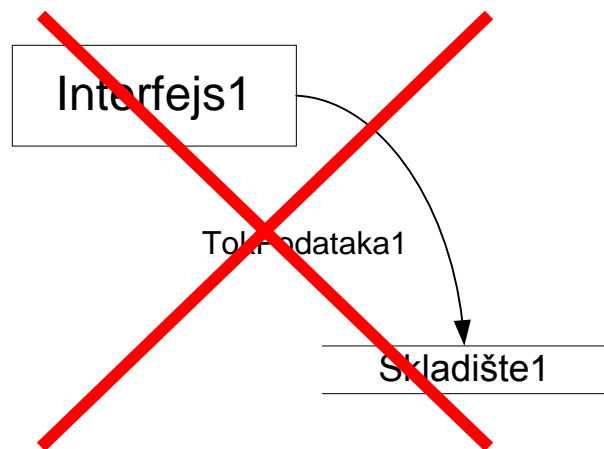
- Било која ДТП компонента може бити почетак или крај тока под условом:

Токовима података се не могу непосредно повезати два складишта, два интерфејса или складише и интерфејс

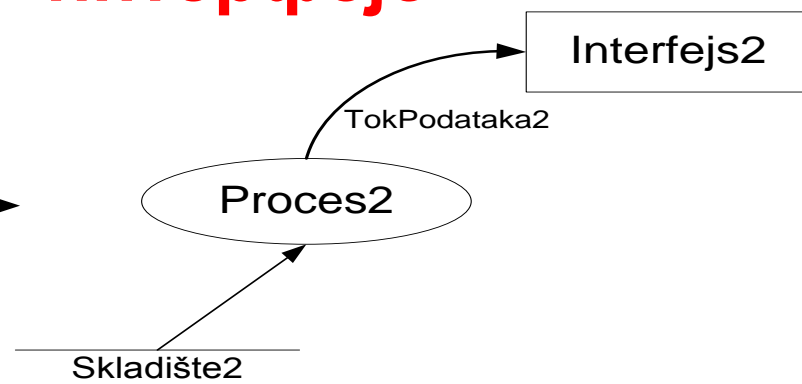
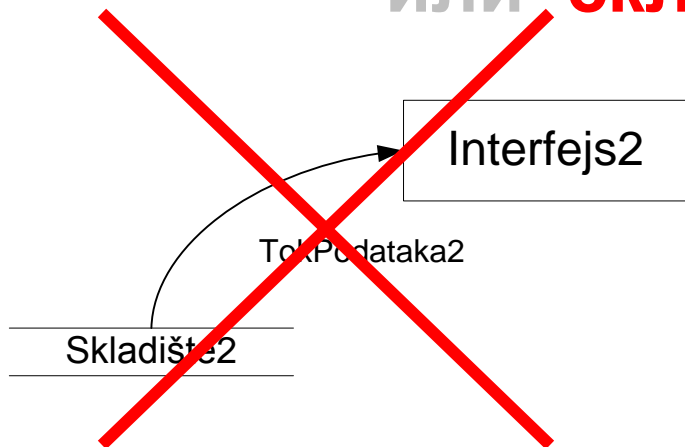


Токовима података се не могу непосредно повезати два складишта, два интерфејса или складише и интерфејс





Токовима података се не могу непосредно повезати два складишта, два интерфејса или складише и интерфејс

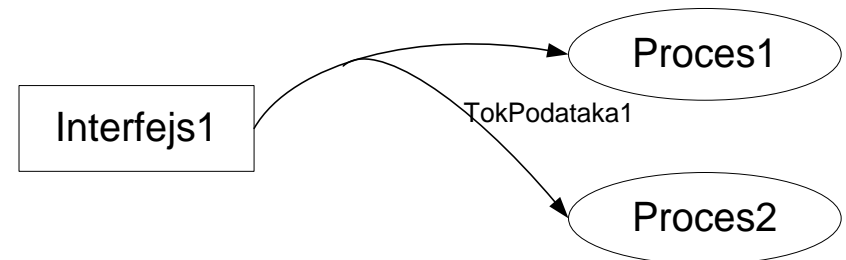
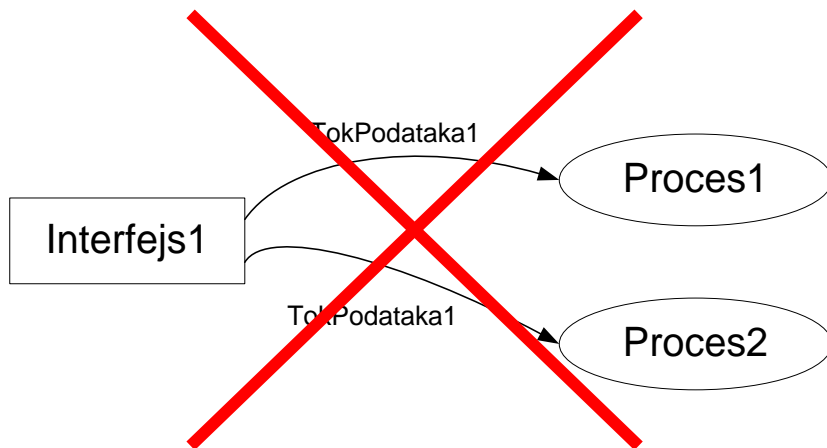


- Сваки ток података на ДТП-а мора имати име, које треба да одражава значење података које он носи.
- Изузетак су токови који иду ка, односно од складишта података а користе целу структуру складишта. Они не морају бити именовани.
- Да би се остварила читљивост и разумљивост ДТП-а, ова имена треба да буду природна, а не нека специфична, претерано скраћена имена.

4. Правило у креирању ДТП-а



- Ток података се може гранати или спајати.
- Истоимени токови на ДТП у суштини претстављају гранање-спајање једног тока са истом структуром података.



- Сваки процес има назив и ознаку.
- Назив процеса треба да прецизно означава функцију коју он обавља.
- Ако пројектант није у стању да додели име процесу то само значи да не разуме функцију коју процес обавља.
- Бројна ознака процеса служи за референцирање процеса, а не представља редослед извршавања процеса.

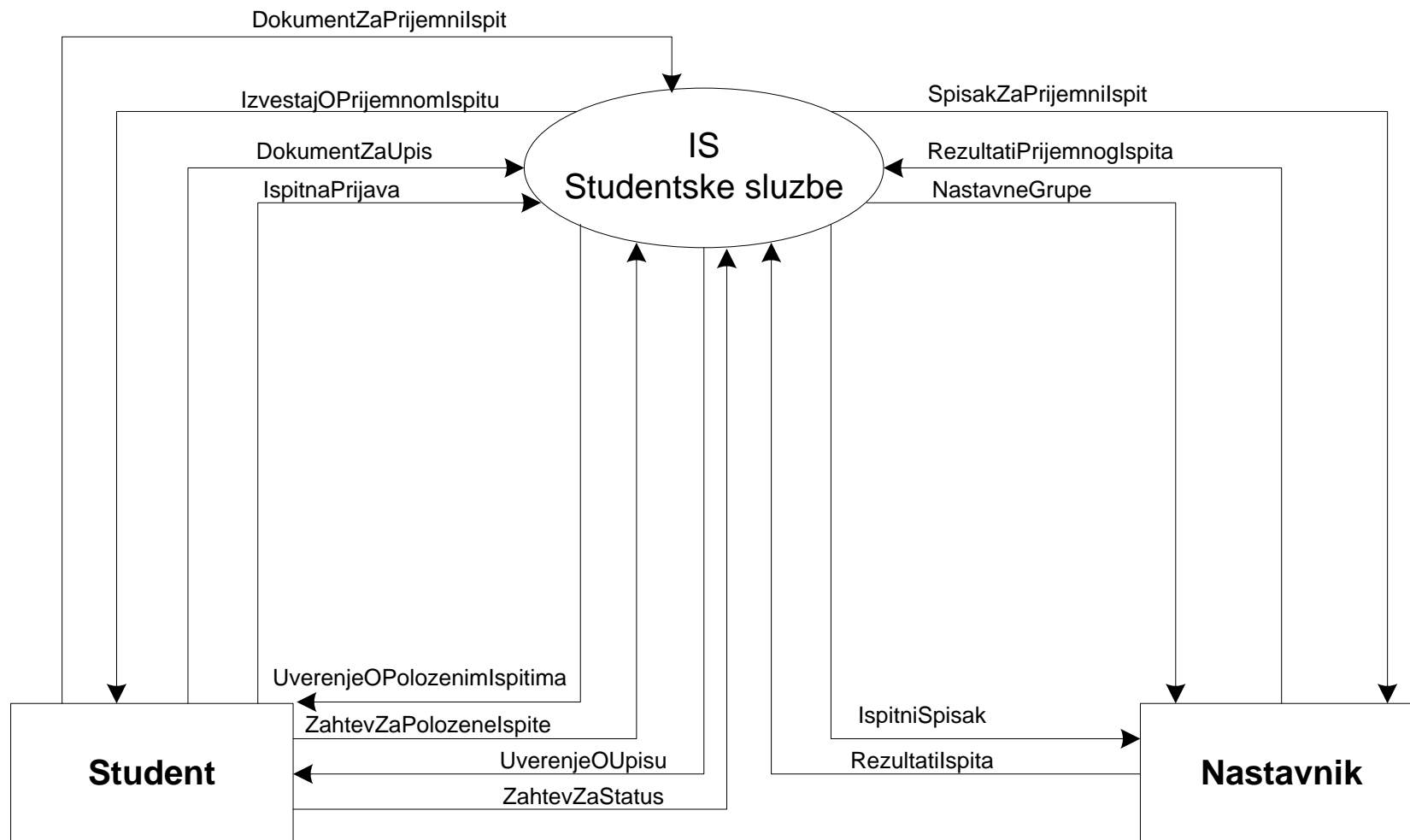
- Сваки процес мора да има барем један улазни и барем један излазни ток података.
- Процес без улазног тока генерисао би излаз ни из чега, а процес без излазног тока је несврсисходан.

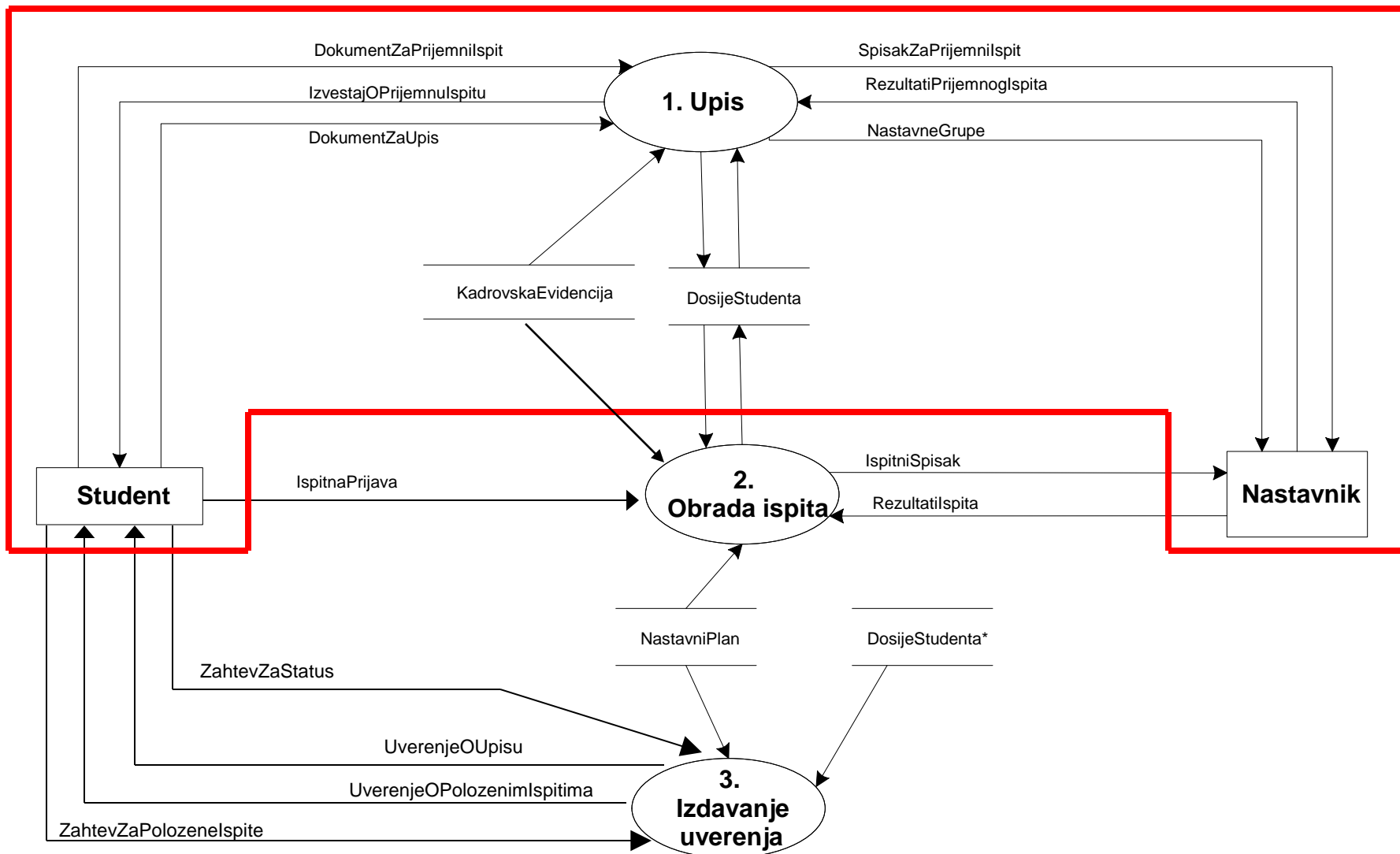
- Свако складиште такође треба да има барем један улазни и барем један излазни ток.
- Дозвољава се да складиште нема улазни ток, подразумевајући да се формира и ажурира у неком другом систему.
- Дозвољава се да складиште нема излазни ток, подразумевајући да посматрани систем формира и ажурира складиште које се користи у неком другом систему.

- Да би се ДТП-ови лакше цртали, односно избегло непотребно пресецање линија, дозвољава се да се једно складиште или интерфејс на једној слици вишеструко понове. Тада се они означавају звездицом.
- Сваки интерфејс мора да има барем један, било улазни, било излазни ток података, иначе би био изолован од система који се моделује.

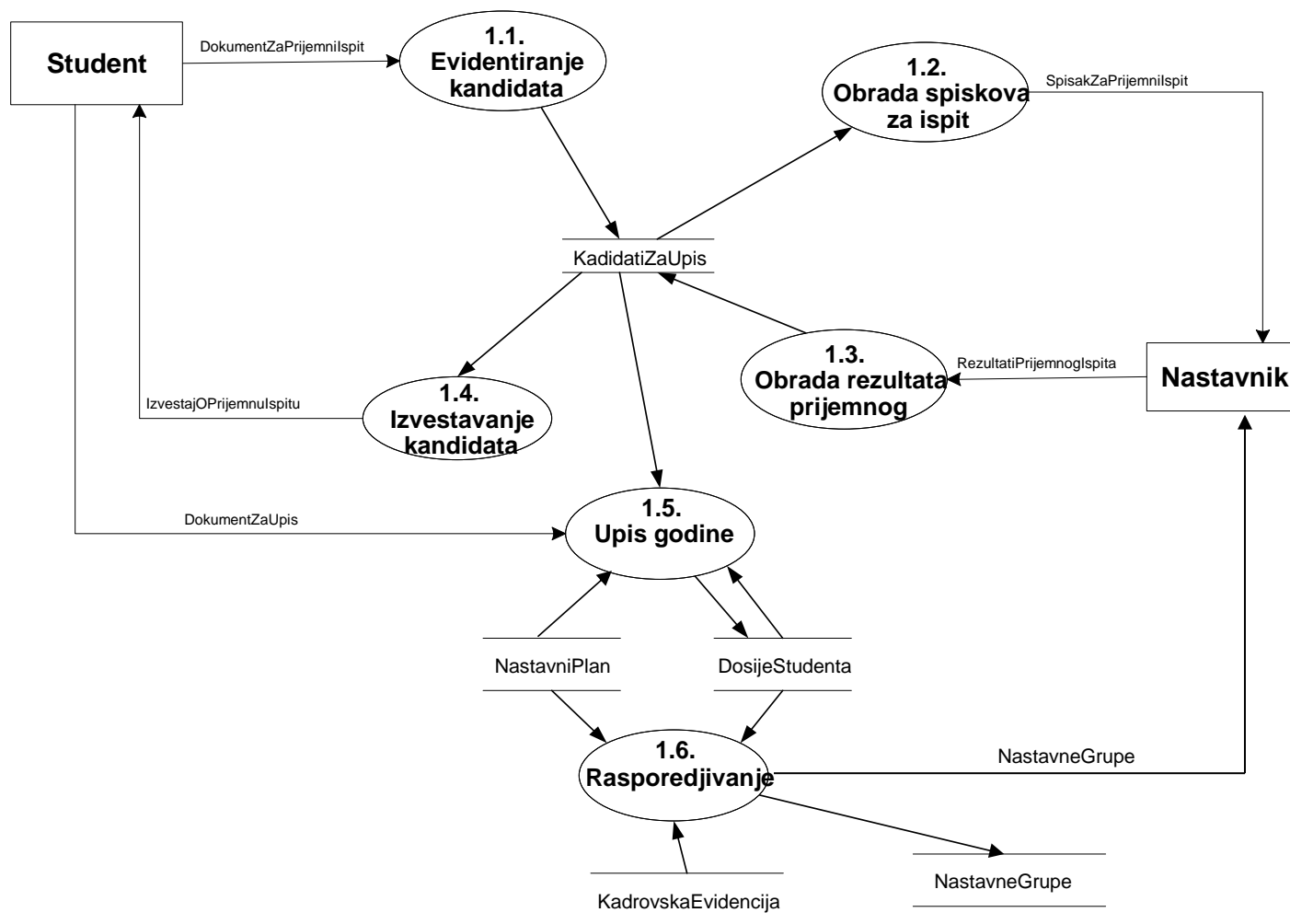
- Један информациони систем може бити веома сложен и може садржати:
 - велики број процеса,
 - токова података,
 - складишта података и
 - спољних објеката.
- Хијерархијска декомпозиција ДТП се изводи на тај начин што се један процес вишег нивоа апстракције декомпонује и приказује помоћу новог целокупног ДТП на нижем нивоу апстракције.

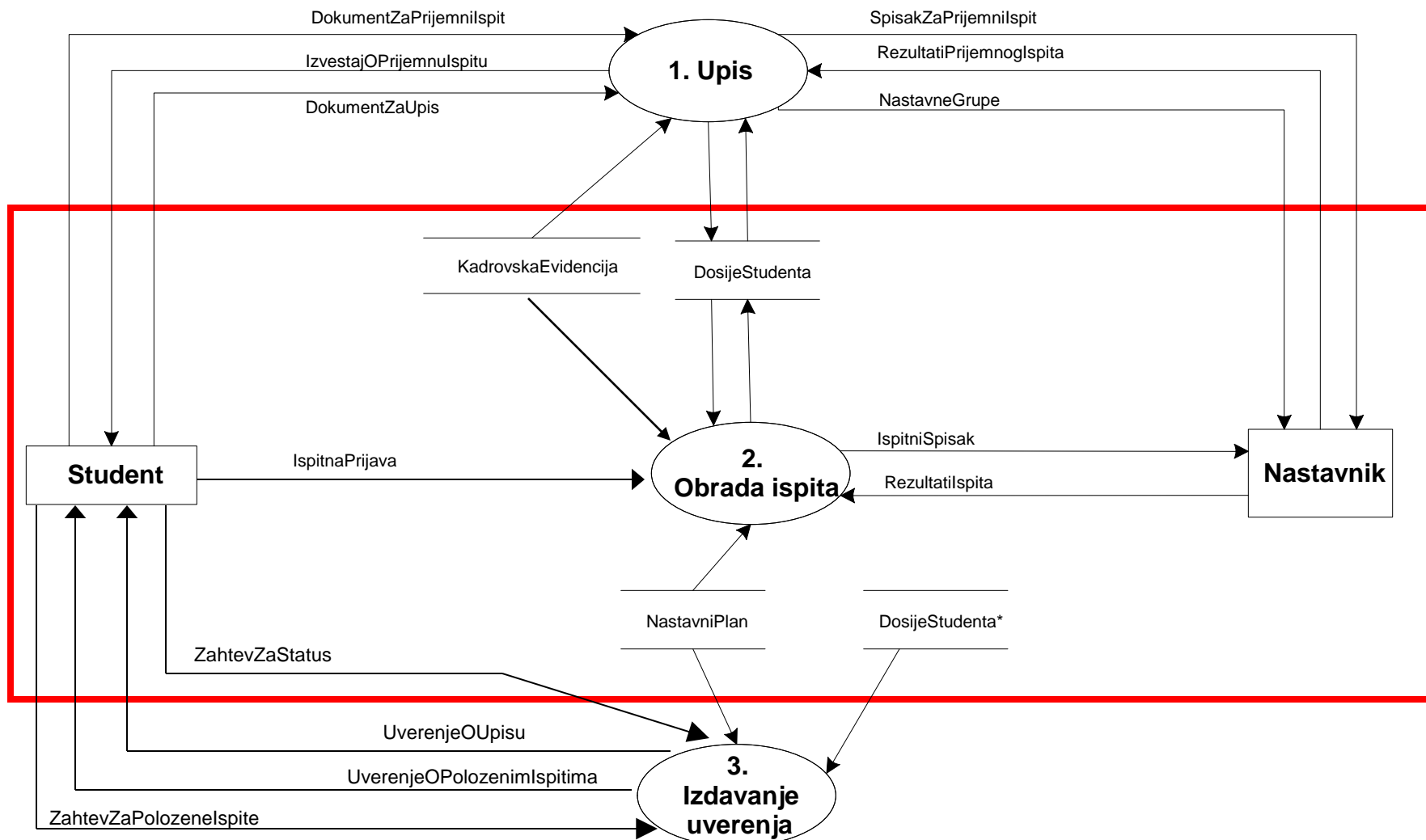
Правилно конструисан ДТП

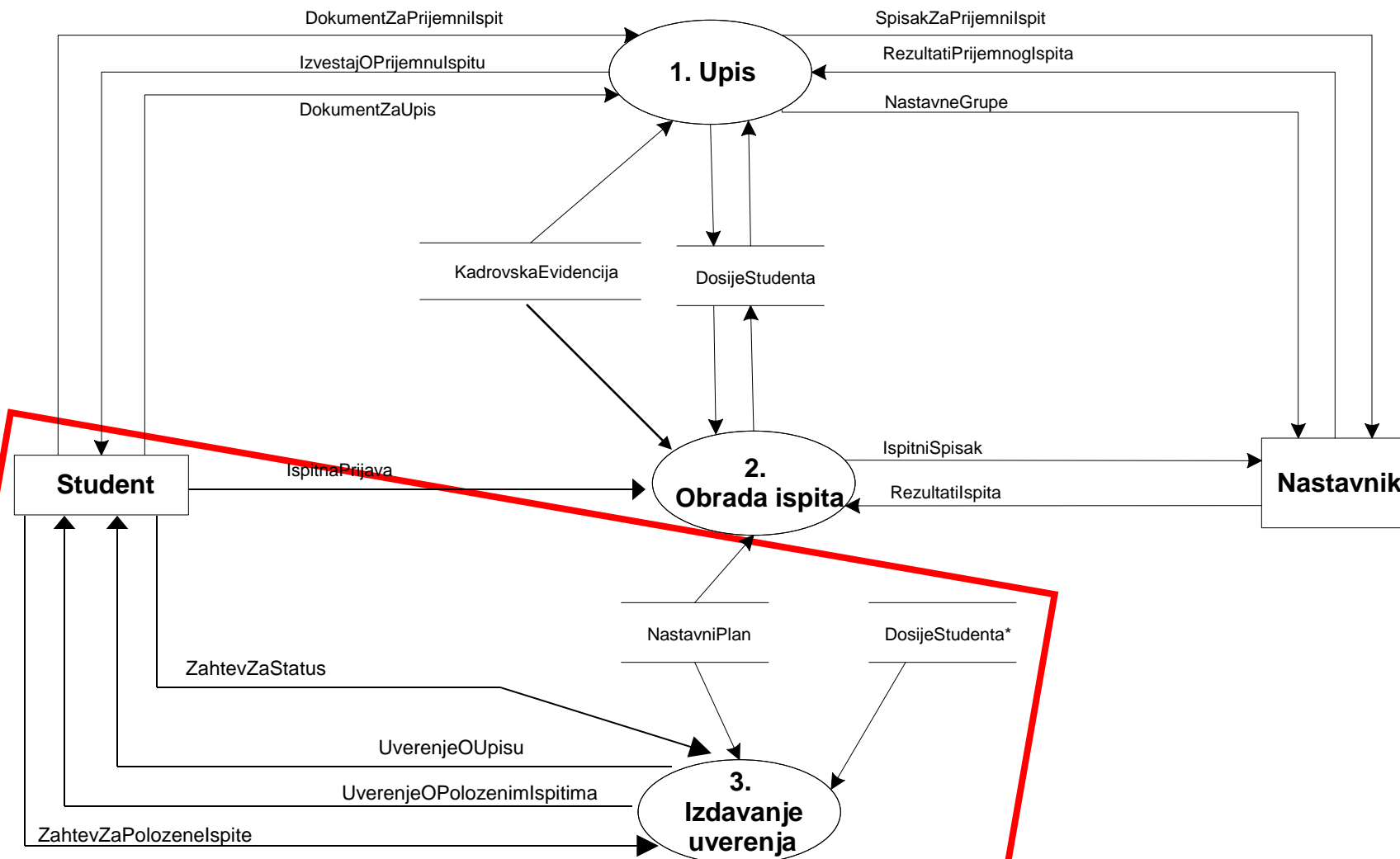




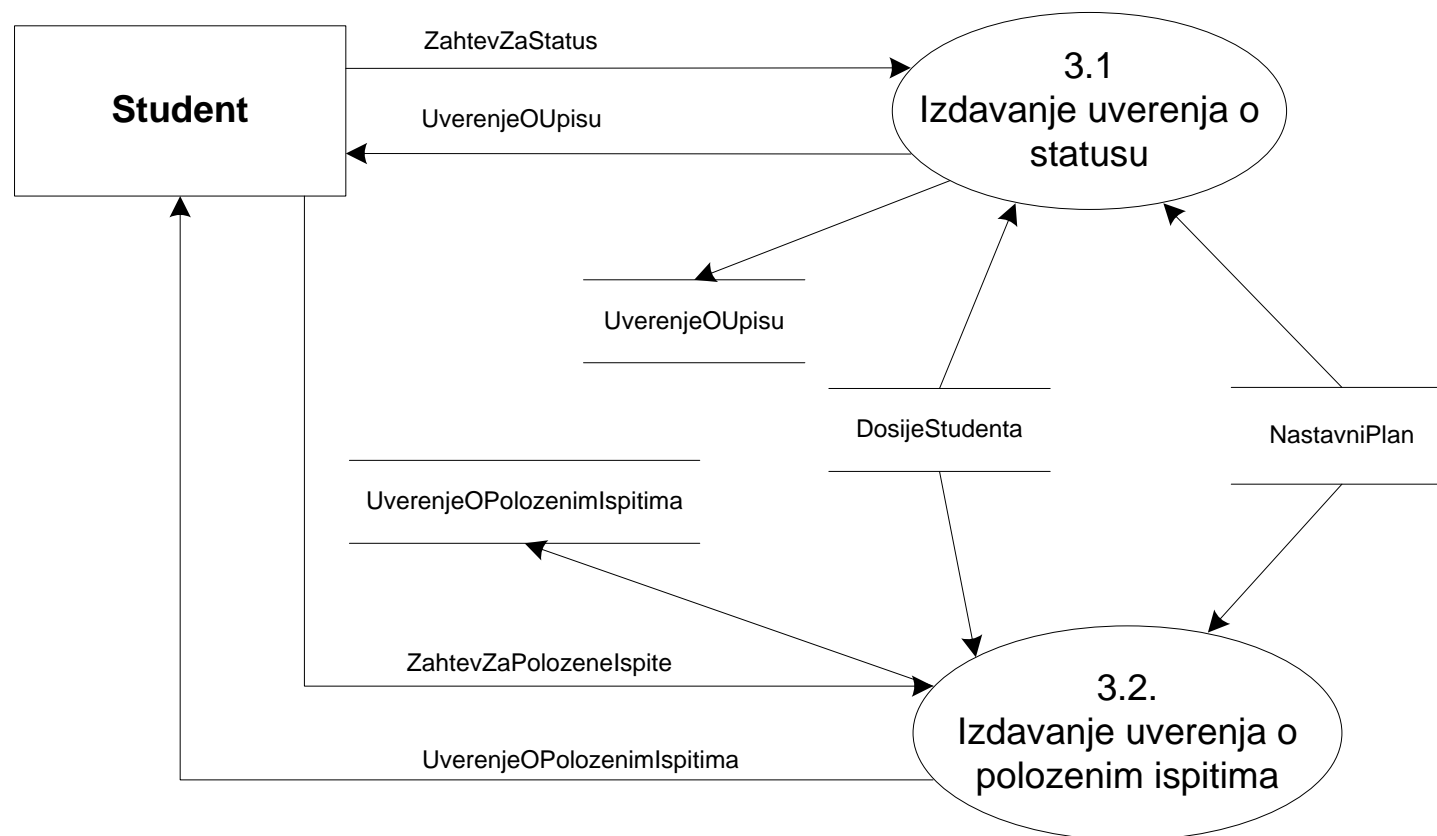
1. Upis







3. Izdavanje uverenja



1. Правило при декомпоновању ДТП-а Ф Н

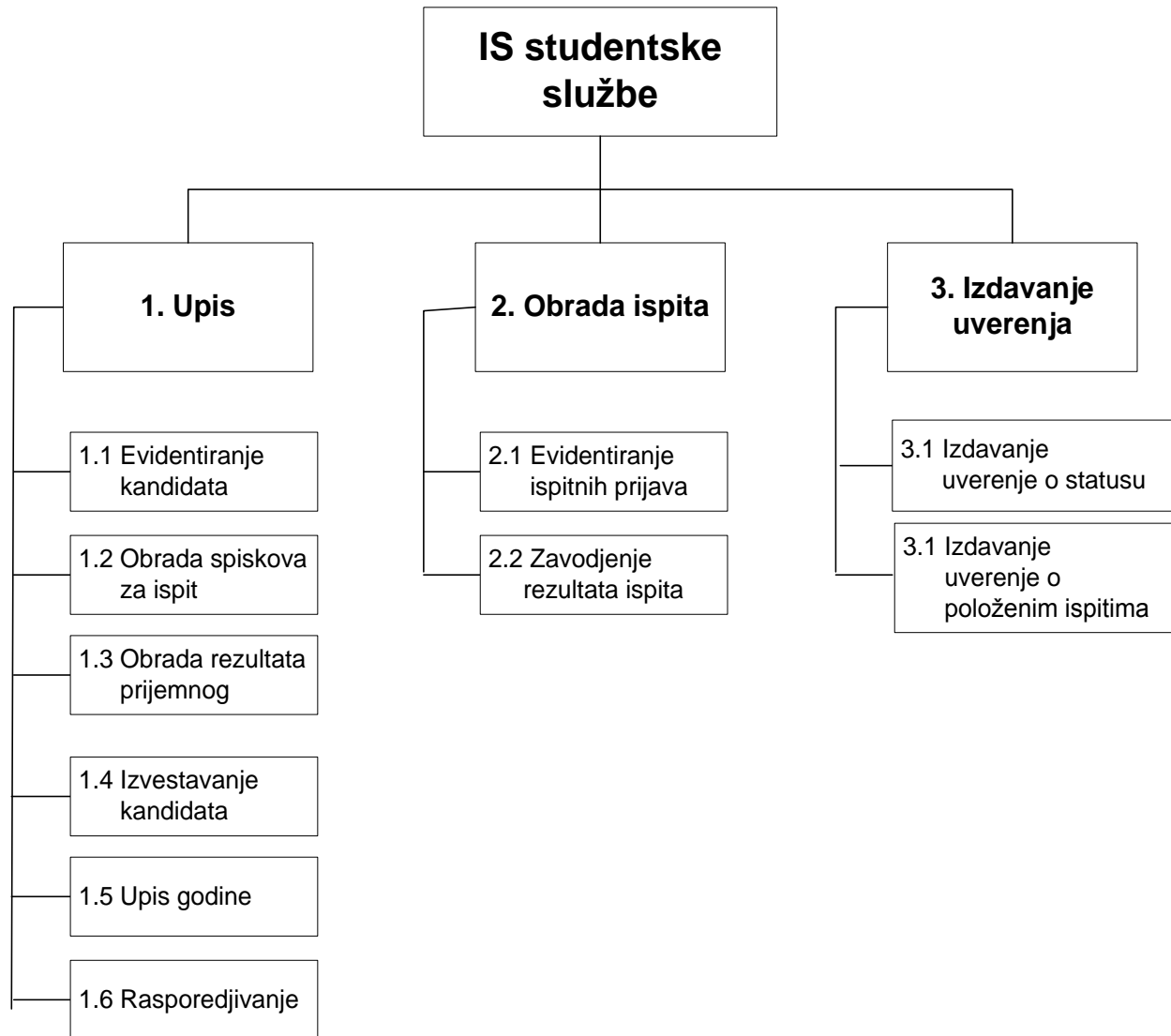
- Дијаграм највишег нивоа, који по правилу садржи само један процес који представља цео ИС, интерфејсе са којима ИС комуницира и одговарајуће токове података назива се **дијаграм контекста**.

2. Правило при декомпоновању ДТП-а Ф Н

- Дијаграм првог нивоа представља декомпозицију дијаграма контекста.
- Процеси на њему означавају се бројевима 1, 2, 3, ...
- Дијаграми нижих нивоа, као целина, су означени са ознаком процеса чије детаље представљају, а процеси на њима повлаче са собом бројну ознаку надређеног процеса, односно посматраног дијаграма.

3. Правило при декомпоновању ДТП-а Ф Н

- Уобичајно је да се у документацији за спецификацију ИС помоћу ССА, целокупан скуп, или неки подскуп хијерархијски декомпонованих дијаграма, представи **дијаграмом декомпозиције**.



4. Правило при декомпоновању ДТП-а Ф Н

- Процеси који се даље не декомпонују се називају примитивни процеси и за њих се даје спецификација логике њиховог одвијања.
- Опис логике примитивних процеса назива се мини-спецификација система.

5. Правило при декомпоновању ДТП-а Ф Н

- Поред процеса, могу се декомпоновати и токови и складишта.
- Декомпозиција токова и складишта се не приказује на ДТП, већ у речнику података.
- Могу се декомпоновати само они токови и складишта који у себи садрже независне компоненте, компоненте чија унија чини ток или складиште које се декомпонује.

6. Правило при декомпоновању ДТП-а Ф Н

- Најзначајније правило које се мора поштовати при декомпозицији процеса је правило баланса токова:
 - Број улазних и излазних токова на ДТП вишег нивоа мора да буде еквивалентан са бројем улазних и излазних токова тога процеса на дијаграму нижег нивоа.
 - Сви токови који улазе, односно излазе из једног процеса, морају се појавити као улазни, односно излазни токови на дијаграму којим је посматрани процес декомпонован.

7. Правило при декомпоновању ДТП-а Ф Н

- Складишта података, представљају стања система, односно фундаменталне, унутрашње карактеристике како целог ИС, тако и сваког појединачног процеса.
- Због тога се она могу појавити и на нижим нивоима декомпозиције, иако се нису појављивала на претходним.
- Међутим, ако се појави на једном нивоу, уз један процес, мора се надаље појављивати на свим нижим нивоима декомпозиције тога процеса.

8. Правило при декомпоновању ДТП-а Ф Н

- Један ДТП, по правилу, треба да садржи 2 - 7 процеса.
- Већи број процеса од овога значио би да је "прескочен" један ниво декомпозиције.
- Процес се даље не декомпонује ако представља примитивну функцију (логичка јединица посла).

МЕТОДОЛОГИЈА МОДЕЛИРАЊА ПРОЦЕСА

- Успешно моделирање захтева, с једне стране, добро познавање реалног система, а са друге добро познавање метода и техника које се користе.
- Поред тога резултат који се добија зависи од:
 - искуства и способности аналитичара,
 - од времена којим се располаже,
 - организованости система који се анализира,
 - комуникације са корисницима система и
 - других сличних фактора.

- Основни циљ методе ССА, налажење логичког модела ИС, може се остварити било "директним моделирањем", на основу познавања суштинских процеса реалног система, било "снимањем", односно извлачењем логичког из постојећег физичког модела ИС.
- У пракси се, наравно, комбинују ова два приступа - полазећи од неког општег "теоријског" модела одређеног типа (врсте) реалног система (производног предузећа, трговине, банке и сл.), анализира се конкретан постојећи систем, да би се утврдиле његове специфичности и посебни захтеви.