ČVUT v Praze
Fakulta dopravní

Návod na zpracování semestrální úlohy z předmětu SYSA

Tento materiál slouží jako návod na vypracování průběžné domácí úlohy z předmětu systémová analýza. Dokument je verzován a průběžně doplňován.

Vypracoval: **Petr Bureš a kol.**
Datum aktualizace: **13. 3. 2019**
Verze: **12**

Historie verzí

verze	datum	komentář
1	28.10.2015	úvodní verze návodu s popisem prvních 2 částí průběžné domácí úlohy.
2	29.10.2015	z požadavků na tabulku vazeb odstraněn sloupec složitost, z tabulky vazeb odstraněn požadavek na směrovost, a požadavek na povinnost parametrů četnost a typ a další drobné korekce.
3	3.11.2015	revidován požadavek na strukturu 1. části, identifikace systému. Doplněny poznámky popisující proč je struktura 1. části úlohy navržena a za jakých podmínek je nadbytečná.
4	4.11.2015	dodán popis 3. části úlohy (regulárnost rozhraní)
5	6.11.2015	úpravy 3. části úlohy, zvýraznění důležitých částí, popsání rozdílu mezi IN a OUT, doplnění příkladu na tabulky parametrů (I=O) a drobné textové korekce
6	24.11.2015	dodána 4. část úlohy
7	30.12.2015	doplněna stručná verze 5. a 6. úlohy
8	9.10.2016	úprava podle nového zadání (z 6 úloh na 5, rozšíření 1. části atp.)
9	25.10.2016	drobné, kosmetické úpravy formálních požadavků a požadavků na vypracování 1 a 2 úlohy
10	15.1.2017	úpravy v části o hierarchické dekompozici. Dodány obrázky hierarchické (HiD) a topologické dekompozice a postuláty. U HiD zdůrazněna možnost neproveditelnosti dekompozice.
11	6. 3. 2019	Milan Sliacky – sjednocení formálních požadavků s videonávody, doplnění textů, informace ke kreslení v MS Word, stylistické úpravy, návod na nalezení genetického kódu
12	13.3.2019	Bureš, Růžička, Bělinová; drobné revize dokumentu.

Obsah

Formální požadavky

Formální požadavky se týkají formy a způsobu odevzdávání domácí úlohy.

Zadání úlohy je zde.

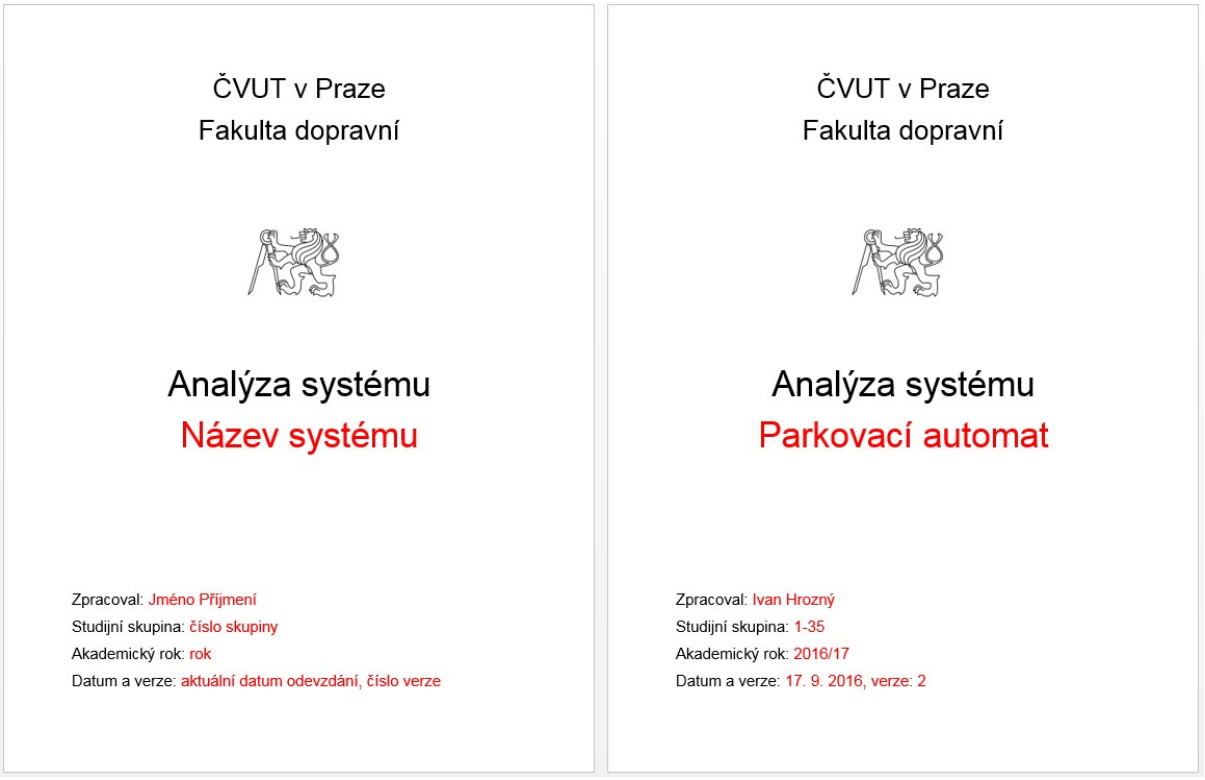
Způsob odevzdávání

- domácí úloha se odevzdává elektronicky, **emailem**
- **předmět** emailu obsahuje identifikaci kurzu, číslo poslední odevzdávané úlohy a její verze, vše je oddělené znakem podtržítka takto: **SYSA_ulohaX_vY,** kde X je číslo poslední odevzdávané úlohy, Y je číslo její verze, každé odevzdání Vaší práce vede k jejímu zvýšení o 1, číslování začíná „1“, příklady:
    - SYSA_uloha1_v1,
    - SYSA_uloha1_v2,
    - SYSA_uloha2_v1, …
- **tělo emailu** ideálně **obsahuje komentář** k verzi, např.: zasílám 2. verzi domácí úlohy obsahující upravenou první část a zároveň druhou část úlohy. …
- **přiložený soubor** je pojmenovaný podobně jako předmět emailu, navíc obsahuje rok kurzu a jméno autora takto: **SYSA_2019_Prijmeni_Jmeno_ulohaX_vY.ZZZ**. Příklady:
    - 1. odevzdání: **SYSA_2019_Berka_Michal_uloha1_v1.doc**
        - obsahuje zpracovanou 1. část úlohy
    - 2. odevzdání: **SYSA_2019_Berka_Michal_uloha1_v2.doc**
        - obsahuje zapracované připomínky k 1. části úlohy, nebo zkrátka student poslal další verzi této části
    - 3. odevzdání: **SYSA_2019_Berka_Michal_uloha2_v1.doc**
        - obsahuje 2. část úlohy, ale samozřejmě (úlohy navazují a odevzdávají se postupně) obsahuje také 1. část úlohy (akceptovanou nebo neakceptovanou)
    - k verzím
        - verze mohou reprezentovat jak nově zpracované části, tak i pouze zapracování oprav, nebo obojí najednou.
        - to co skutečně ve verzi odevzdáváte, popište do těla mailu
- ZZZ znamená příponu souboru, akceptované formáty odevzdávaného souboru jsou **pdf, doc a docx**
- vždy je (emailem) odevzdáván pouze **JEDEN soubor** (zahrnující vše předešlé), nejsou akceptovány maily obsahující více příloh, například obrázky, tabulky apod.

Formální úprava dokumentu

- dokument obsahuje úvodní identifikační stránku (viz obrázek 1) s těmito informacemi:
    - **název systému**

- identifikace studenta, skupiny a akademického roku
- datum odevzdání a číslo verze



ČVUT v Praze
Fakulta dopravní

Analýza systému
Název systému

Zpracoval: Jméno Příjmení
Studijní skupina: číslo skupiny
Akademický rok: rok
Datum a verze: aktuální datum odevzdání, číslo verze

ČVUT v Praze
Fakulta dopravní

Analýza systému
Parkovací automat

Zpracoval: Ivan Hrozný
Studijní skupina: 1-35
Akademický rok: 2016/17
Datum a verze: 17. 9. 2016, verze: 2

Obr. 1 Ukázka přední stránky odevzdávané práce, červeně jsou uvedeny požadované položky, úprava přední stránky (loga, rozmístění a velikost textů) je na Vás.

- dokument má obsah (automaticky) generovaný z nadpisů
- dokument může (nemusí) obsahovat historii verzí (viz tento návod)
- dokument je strukturovaný podle částí úloh do **5 číslovaných** kapitol:
  - Část 1 – volba systému a jeho úvodní popis
  - Část 2 – strukturní identifikace systému
  - Část 3 – úloha o společném rozhraní na identifikovaném systému
  - Část 4 – dekompozice na identifikovaném systému
  - Část 5 – chování identifikovaného systému a jeho genetický kód
- kapitoly budou doplňovány v odevzdávaném dokumentu podle termínu odevzdání souvisejících částí úlohy
- nadpisy Obsah a Historie verzí jsou bez uvozující číslovky, číslování tedy začíná kapitolou 1 Volba systému a jeho úvodní popis
- každá kapitola je ideálně dále strukturována do číslovaných nadpisů podle požadavků v zadání úlohy, např. kapitola 1 Volba systému a jeho úvodní popis
  - by mohla být strukturována následovně:
    - 1.1 Název a stručný slovní popis systému

        - 1.2 Slovní popis fungování systému, jeho částí, ...
        - 1.3 Popis okolí systému a jeho vztahu k systému
        - 1.4 Seznam prvků systému
        - 1.5 Grafické schéma systému obsahující prvky a vazby
    - nebo by mohla být strukturována:
        - popis systému a jeho chování
        - výčet prvků systému

- obsažené obrázky, tabulky, schémata a grafy **budou opatřeny číslovaným titulkem** (viz tento dokument). Titulek k obrázku pod obrázkem, např. Obr. 1 Úvodní náhled systému, titulek k tabulce nad tabulkou, např. Tab. 1 Tabulka prvků systému, schémata a grafy titulkovat jako obrázky.

Požadavky na úpravu textu

- v textu dokumentu je použit jednotný font a jednotná velikost, například “Arial 11” (podobné Calibri / Verdana), či “Times New Roman 12”
- Všechny nadpisy mají shodný font (lišící se od fontu textu), úrovně nadpisů se liší (číslováním) velikostí použitého fontu a mezery nad a pod nadpisem
- řádkování v dokumentu 1 až max 1.2
- odstavce jsou odděleny větší mezerou, buď automatickou mezerou mezi odstavci, nebo manuálním odřádkováním
- dokument nemá zbytečné mezery (volné stránky, půlstránky), text jednotlivých kapitol a podkapitol na sebe navazuje.

Požadavky na věcný obsah dokumentu

Věcný obsah se vždy odvíjí od zadání, je tedy nezbytné je mít podrobně nastudované.

Zadání úlohy je zde.

Část 1 Volba systému a jeho úvodní popis

Obecně

- tato část je nejdůležitější protože špatně identifikovaný systém je následně obtížné posuzovat v dalších úlohách.
- je **nezbytné** rozpoznat a popsat strukturu systému v této úrovni:
    - prvky systému (v rozsahu 10-15)
    - prvky okolí
    - (vazby mezi prvky systému vzájemně a prvky okolí)
- je **nezbytné** rozpoznat a popsat **základní** činnosti (procesy) v systému, tak aby bylo zjevné jaké prvky okolí procesy "spouští" a jaké prvky systému jsou v procesech aktivovány.

Jak identifikovat prvky

Úroveň rozlišení identifikovaných prvků

Prvky systému by měly být na přibližně stejné úrovni složitosti / významu, u prvků okolí nevadí i když mají velmi různou úroveň (většinou více obecnou). Prvky uvnitř systému mohou variovat v úrovni složitosti, to ale nemusí být v rámci dané rozlišovací úrovně vadit, snažíme se nalézt takové prvky, které jsou na dané úrovni podrobnosti funkčně důležité.

- DOBŘE/ŠPATNĚ: systém obsahuje prvky, které se fyzicky nacházejí uvnitř jiných prvků. Zde záleží na použití systému, tedy na tom, co chceme systémem modelovat. Pokud modelujeme konstrukci, tedy jak jsou prvky fyzicky pospojovány, jak se například přenáší působení síly, je to v většinou akceptovatelné. Pokud ale modelujeme systémem funkci, tak je většinou zapouzdřující (konstrukční) prvek navíc.
    - **PŘ:** systém PARKOVACÍ AUTOMAT, popisuje funkci (obsloužení řidiče, vydání lístku atp.) a obsahuje prvky *kryt*, *displej*, *tiskárna* a další. Prvek *kryt* je tam navíc nemá žádné funkční opodstatnění.
    - **PŘ:** systém VOZIDLO, popisuje konstrukci (jak jsou jednotlivé prvky konstrukčně uspořádány, propojeny) a obsahuje prvky *karosérie*, *rám*, *motor*, *sedačky* a další. Prvek *karoserie*, ačkoliv zahrnuje (fyzicky obsahuje) další prvky vyjmenované v rozpoznaném systému má zde opodstatnění, existují fyzická propojení mezi karosérií a dalšími (obsaženými) prvky, které chci popsat.
- ŠPATNĚ: systém obsahuje prvky výrazně složitější a na dané úrovni podrobnosti dekomponovatelné a zároveň prvky jednoduché nacházející se na dané úrovni podrobnosti. například:

    - **PŘ:** systém KYTARA, obsahuje prvky *kytara*, *kabel*, *reproduktory*, a další. Prvek *kytara* by měl být reprezentován více prvky
    - **PŘ:** systém POČÍTAČ, obsahuje prvky *PC*, *myš*, *klávesnici*, *monitor*, a další hromadu periferií. Prvek *PC* by měl být reprezentován více prvky
- ŠPATNĚ: systém obsahuje zároveň prvky na vyšší a nižší rozlišovací úrovni, kdy prvek na vyšší rozlišovací úrovni (obecnější) implicitně zahrnuje prvek na nižší úrovni. např.:
    - **PŘ:** systém SEDAČKA obsahuje prvek *sedačka* a prvek *područka* a další. Prvek *sedačka* by měl být dekomponován na prvky na úrovni podrobnosti dalších prvků (*sedačka* => *sedák*, *opěrák*, *područky*, *skladovací prostor*, atp.) a duplicity po dekompozici odstraněny.
    - **PŘ:** systém POČÍTAČ, obsahuje prvky *HW počítače*, *procesor* a další. Prvek *HW počítače* by měl být dekomponován na prvky na úrovni podrobnosti dalších prvků (*HW počítače* => *procesor*, *paměť*, *základní deska*, *zdroj*, *grafická karta*, atp.) a duplicity po dekompozici odstraněny.
    - **PŘ:** systém NÁDRAŽÍ, obsahuje prvky *hala*, *wc*, *výdejní místo*, *úschovna* a další. Prvek *hala* by měl být dekomponován na prvky na úrovni podrobnosti dalších prvků (*hala* => *wc*, *občerstvení*, *úschovna*, *výdejní místo*, atp.) a duplicity po dekompozici odstraněny.

Typ identifikovaných prvků

Prvky systému by měly být přibližně stejného typu. To neznamená, že by to měly být stejné resp. homogenní prvky, viz příklad se zastávkami, ale nemělo by nastat **kombinování různých základních typů prvků:**

- organických a mechanických,
- pohyblivých a statických,
- hmotných a nehmotných,
- stavových (existujících v určitém čase = zachycujících určitý krátkodobý stav) a nestavových (fyzických prvků)

je problematické a je dobré se tomuto zdaleka vyhnout. Zavádí to do systému problémy, které vyniknou zejména při popisu jeho chování. Příklady:

- ŠPATNĚ: systém obsahuje zároveň prvky statické a prvky “putovní”, např. prvky které se volitelně vyskytují v dalších prvcích systému. Takový systém lze obtížně charakterizovat v procesech, které zahrnují pohyb “putovního” prvku po prvcích statických. Je nejlepší se “putovnímu” prvku úplně vyhnout a brát jej jen jako “něco”, co “putuje” po vazbách a je tedy implicitně přítomno. Příklady:
    - **PŘ:** systém NÁDRAŽÍ obsahuje prvky *čekárna*, *úklid* (místnost), *kasa* (místnost), *prodavač*, *uklízečka*, *cestující* a další. Prvky *prodavač* / *uklízečka* / *cestující* jsou spíše prvky okolí, které vstupují do systému do prvků *kasa* / *úklid* a poté se systémem pohybují a tím aktivují další prvky. Samostatně by v systému neměly existovat (jejich pohyb by se modeloval velmi obtížně)
    - **PŘ:** systém PISTOLE obsahuje prvky *rukojeť*, *zásobník*, *kohoutek*, *náboj* (střela+nábojnice) a další. Prvek *náboj* je prvek okolí, který vstupuje do

zásobníku a poté putuje zbraní podle toho, co se s ní dělá, (natažení, vyprázdnění komory, výstřel, apod). *náboj*, který se v průběhu rozpadá na nábojnici a střelu se systémem pohybuje a tím aktivuje další prvky!! samostatně by v systému neměl existovat (jeho pohyb by se modeloval velmi obtížně)

- **PŘ:** systém MOTOR obsahuje prvky *píst*, *komora*, *zážeh*, *expanze*, *komprese* a další. Jedná se o míchání popisu systému typu proces, kdy prvky nereprezentují fyzické části popisovaného objektu, ale určité stavy objektu / jeho částí, v čase a systému čistě fyzického, kdy jednotlivé jeho části sice v čase vykonávají různé funkce, ale na struktuře systému se to nijak neprojeví. Tyto dva "náhledy" na popisovaný objekt jsou většinou neslučitelné.

Výstupy této části

Výstupy této části úlohy budou tyto kapitoly:

- Název a stručný slovní popis systému
- Slovní popis fungování systému, jeho částí, odůvodnění volby rozlišovací úrovně a způsobu pohledu na systém
- Popis okolí systému a jeho vztahu k systému
- Seznam prvků systému
- Grafické schéma systému obsahující prvky a vazby

POZN: Výše uvedená struktura umožňuje zjistit*, že v úvodní identifikaci nic neschází*. Pokud bude výstup této části strukturován jinak, například bude obsahovat souvislý text, ale bude z něj jasné, že pozorovatel systém uchopil dobře, tj.: bude dobře popisovat alternativy chování a zapojení prvků okolí a prvků uvnitř systému, díky popisu chování bude implicitně obsahovat, co vlastně prvky dělají a jaké je jejich propojení, PAK je struktura uvedená výše nadbytečná.

1.1 Název a stručný slovní popis systému

- tato část obsahuje název a zároveň abstrakt tedy 2-3 řádkové shrnutí co vlastně systémem popisujete. Zde můžete být i subjektivní, tedy stručně napsat i proč.
    - jaký objekt reálného světa (dále i objekt RS) popisujeme
    - abstrakt

1.2 Rozšířený slovní popis

- **rozsah**: odstavec, ne více než půl stránky
- V této části popíšete **fungování systému, jeho částí**, odůvodníte volbu rozlišovací úrovně a způsob pohledu na systém.
- Jedná se o několik odstavců souvislého textu, ze kterého vyplývá (odrážky níže nereprezentují odstavce):
    - co chcete systémem na objektu zachytit / popsat (**jaká je jeho funkce**) a jaká je funkce jeho částí (prvků). Tento popis je volný text obsahující názvy částí systému "propojené" činnostmi tak aby bylo zřejmé, co se v systému děje.

    - u částí (prvků) mějte na paměti jejich přibližně stejnou složitost (hloubku zanoření při popisu) to Vám určuje rozlišovací úroveň, kterou slovně určíte.
    - z popisu funkce také vyplývá pohled na systém.
- CO SEM NEPATŘÍ:
    - obecný popis typu, “protože objekt RS znám tak jej popisuji ...”, “systém uspokojuje potřeby zákazníků / uživatelů / atp. …”
- CO SEM PATŘÍ
    - buďte konkrétní, popisujte konkrétní systém, u pohledu na objekt RS, tedy co chcete hlavně systémem modelovat.
    - Buďte konkrétní, nepoužívejte obecné nic neříkající fráze. **Každá věta by měla obsahovat užitečnou informaci s ohledem na Vámi popisovaný objekt RS**.

1.3 Popis okolí systému a jeho vztahu k systému

- **rozsah**: odstavec, ne více než několik řádků
- jak je systém **zasazen do svého okolí**, tedy jaké jsou základní části **uvnitř** systému a jaké jsou části **mimo** systém (tedy okolí), zde v této části se jedná hlavně o uchopení **funkce systému**.
- jaká jsou základní chování objektu, opět se **nejedná** o celkový výčet všech možných chování / procesů, ale z popisu by mělo vyplývat **jak (kdy/proč)** vámi identifikované, části okolí působí na systém (jeho části).

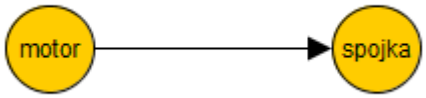
1.4 Seznam prvků systému a okolí

- **rozsah**: výčet, cca 1 strana, ke každému prvku okolí a systému 1 odrážka s doplňujícím textem
- **výčet všech prvků systému**, kde u každého prvku je zmínka o tom co dělá, jaká je jeho **vstupně-výstupní funkce**. Například:
    - **PŘ:** systém VOZIDLO s několika prvky, z nichž jeden je “motor”
      motor - mění energii (v palivu) na rotační pohyb,
    - **PŘ:** systém POČÍTAČ s několika prvky, z nichž jeden je “grafická karta”
      grafická karta - vykresluje obraz dle dodaných modelů, textur a posílá jej do monitoru,
- **výčet všech prvků okolí**, kde u každého prvku je zmínka o tom co dělá, jak ovlivňuje systém. Například:
    - **PŘ:** systém VOZIDLO s několika prvky okolí, z nichž jeden je “vozovka”
      vozovka - vlastnosti vozovky ovlivňují kvalitu jízdy vozidla
    - **PŘ:** systém POČÍTAČ s několika prvky okolí z nichž jeden je “el. přípojka”
      el. přípojka - napájí zdroj počítače síťovým napětím 230 V
- **POZN**: pokud rozšíříte popis chování (části 1.2 a 1.3) tak, že obsahuje podrobnější popis různě aktivovaných procesů prostřednictvím prvků a jejich zapojení v nich (co dělají, co si předávají), tak splníte požadavek hlavní této části (popsat funkci

prvků) a není je zde potřeba znovu uvádět (jednalo by se o duplicitu). Pouze uvedete výčet prvků systému a jeho okolí, které jste použili v popisu chování.

1.5 Grafické schéma systému obsahující prvky a vazby

- tato část obsahuje orientovaný graf vyjadřující všechny identifikované prvky systému i okolí propojené všemi identifikovanými vazbami.
- prvky okolí musí být v grafu odlišené od ostatních prvků systému
- prvky musí být pojmenovány (číslem, zkratkou, ideálně: názvem) a tato identifikace se musí objevit i v tabulce prvků. Vazby být pojmenovány (očíslovány) nemusí, jejich identifikace je dána prvky, které propojují.
- vazby jsou primárně orientované a vždy musí propojovat 2 prvky, vazbu nelze vést od prvku ke “skupině” několika prvků. Pokud existuje obousměrná vazba, je tato vyjádřena buď dvěma jednosměrně orientovanými “propojeními” (vyjádřeno šipkou) nebo jedním obousměrným.



Obr. 2 ukázka vazby mezi prvky motor a spojka

- pokud použijete “malovátko” ve Wordu, používejte kreslící plátno, aby Vám „držely“ vazby s prvky. Můžete použít jakýkoli rozumný editor, například editor yEd (http://www.yworks.com/en/products/yfiles/yed). Nebo můžete **graf nakreslit ručně a výsledek naskenujte** a vložte do dokumentu, platí za podmínky, že to zvládnete udělat v dostatečné kvalitě, nejsou akceptovány žádné čmáranice. Nevýhodou ručního kreslení je, že rozmístění prvků nemusí být na první pokus optimální, což vede k tomu, že vazby mezi nimi budou nepřehledné. Pak důrazně doporučujeme obrázek překreslit a rozmístění prvků optimalizovat.

Část 2 Strukturní identifikace

Obecně

- tato část již obsahuje pouze formalizovaný zápis předchozí identifikace.
- její součástí je tabulka prvků, tabulka vazeb, a matice sousednosti. Doporučujeme uvádět v tomto pořadí.

Výstupy této části

2.1 Tabulka prvků

- tato část obsahuje formalizovanou tabulku popisující obecně vlastnosti prvků systému.
- VOLITELNĚ: do tabulky prvků systému můžete (ale nemusíte) zapsat také prvky okolí, ty potom identifikujete pomocí sloupce/klasifikace "významnost" jako okolí/vnější
- tabulka prvků **obsahuje povinně tyto sloupce**:
    - *číslo* (identifikace prvku) – značka, kterou je prvek označen v grafu systému, nebo pořadové číslo prvku.
    - *název* – plný nezkrácený název prvku
    - *funkce* – jedno maximálně dvou slovní extrakt z neformálního popisu prvku zachycující hlavní funkci prvku v systém. Je pravděpodobné, že více prvků bude mít díky této generalizaci stejnou funkci.
      Např.: grafická karta .. vykreslovací, spojka .. spojovací/rozpojovací, atp.
    - *významnost prvku* – uveďte, které prvky jsou vnitřní / hraniční (vstupní / výstupní), vnější, řídicí / podřízené, …,
    - *vlastnosti / parametry* – vlastnosti prvků jsou generalizované, tak aby je měly téměř všechny prvky (barva, váha, cena, napájení, spolehlivost, složitost, atp.) pokud daný prvek vlastnost nemá, je položka vlastnosti ponechána prázdná, případně proškrtnutá. Vlastnosti volíte sami na základě vaší znalosti systému, minimálně 2, tabulka tedy bude obsahovat minimálně 2 sloupce vlastností.
- Konkrétní vlastnosti prvků, které již nejde zobecnit tak, aby platily pro více vazeb, **nemají** v přehledové tabulce smysl. Např.:
    - **PŘ:** systém POČÍTAČ: na vazbě mezi prvky *monitor – uživatel* mohou existovat vlastnosti / parametry *rozlišení*, *jas*, *kontrast*, které ale pro jiné vazby např. *základní deska – paměť* nemají smysl.

Tab. 1 Příklad tabulky prvků pro systém POČÍTAČ (poslední 3 sloupce jsou obecné parametry a jsou závislé na vámi zvoleném systému a Vaší imaginaci)

číslo	název	funkce	významnost	složitost	cena	spotřeba	záruka

1	grafická karta	vykreslovací	vnitřní	5	5000 Kč	60 W	2 roky
2	zákl. deska	propojovací	vnitřní	5	2500 Kč	15 W	2 roky

2.2 Tabulka vazeb

- tato část obsahuje výčet všech vazeb mezi prvky a **jejich vlastností** ve vámi identifikovaném systému mezi sebou a také mezi prvky okolí (vazby prvků systému s okolím uvádějte v samostatné tabulce), kde **každá vazba je jeden řádek tabulky**.
- tabulka vazeb obsahuje povinně tyto sloupce:
    - *číslo* (identifikace vazby) – pořadové číslo vazby (tímto číslem může být vazba označena v grafu).
    - *zdroj* – plný nezkrácený název prvku, ze kterého vazba vychází
    - *cíl* – plný nezkrácený název prvku, do kterého vazba vchází
    - *vlastnosti / parametry* – tabulka vazeb by mohla mít mnoho parametrů, ale pro jednoduchost požadujeme alespoň dva. Dále uvádíme příklad univerzálních parametrů použitelných pro každé rozhraní:
        - *četnost* – vyjadřuje jak často / kdy se vazba aktivuje, její hodnoty mohou nabývat různých klasifikací od “trvale” přes “pravidelně 1 Hz”, “nahodile”, “na výzvu”, až po “vůbec” - po tom by ale vazba neexistovala - klasifikace záleží na představivosti pozorovatele objektu RS
        - *typ/způsob realizace* - vyjadřuje jak je vazba uskutečněna, může jít o mechanické propojení, přenos informace, přenos energie, datové propojení - klasifikace záleží na představivosti pozorovatele objektu RS
        - *dynamika* – vyjadřuje jestli jde o vazbu v čase stálou či realizující se jen v určitých okamžicích: stálá, vyvolaná událostí ...
    - pokud naleznete jiné “generické” či konkrétní parametry vazeb, které většina vazeb sdílí, tak je zde uveďte, naopak, ne ve všech případech jsou vlastnosti četnost a typ jsou použitelné.

Tab. 2 Příklad tabulky vazeb pro systém POČÍTAČ

číslo	zdroj	cíl	typ vazby	četnost
1	grafická karta	základní deska	informační	2 GHz (rychlost komunikace s procesorem)
2	grafická karta	monitor	informační	60 Hz (zobrazovací frekvence)

2.3 Matice sousednosti

- tato část obsahuje formalizovaný zápis propojení systému pomocí matice sousednosti Spp (čtvercová matice prvek – prvek) = doporučujeme tuto místo jedné z ostatních matic (Spv, Svp, Svv)
- matice sousednosti obsahuje informace o tom, jestli mezi 2 prvky systému existuje vazba a v jakém směru. Pokud je na spojnici mezi prvkem A v řádku a prvkem B ve sloupci 1 pak existuje mezi prvky A a B propojení ve směru A->B.
- pokud je zároveň číslo 1 na spojnici mezi prvkem B v řádku a prvkem A ve sloupci číslo 1 existuje mezi prvky A a B obousměrné propojení (zrcadlení vůči hlavní diagonále).
- prvky mezi kterými propojení (vazba) neexistuje, mají spojnici prázdnou.
- do matice sousednosti Spp **nepatří** prvky okolí.

Příklad matice sousednosti pro systém POČÍTAČ, ilustruje jednosměrné propojení (vazbu) mezi prvkem základní deska (zdroj) a prvkem grafická karta (cíl).

Tab. 3 Ukázka matice sousednosti pro systém POČÍTAČ

Spp	...	grafická karta	základní deska	...
...				
grafická karta				
základní deska		1		
...				

Část 3 Úloha o společném rozhraní

Obecně

- tato část obsahuje popis DVOU vybraných vazeb, do velké úrovně podrobnosti, částečně vychází z předchozí identifikace parametrů systému, ale jde s poznáním hlouběji. Většinou nestačí na této úrovni pracovat s vlastnostmi / parametry identifikovanými na celkové (systémové) úrovni.
- její součástí je slovní popis vazby, popis parametrů a jejich hodnot, vstupní a výstupní tabulka parametrů, podmínka regularity, matice neregularit a popis odstranění neregularity (vyladění vazby)

Jak identifikovat parametry a jejich hodnoty

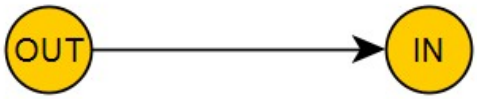
- parametry prvků (a tedy co posílají do resp. očekávají z vazby), na systémové úrovni jsou většinou hodně obecné (výjimky jsou specializované systémy, kde jsou shodné prvky), protože každý prvek má (na podrobnější než obecné rozlišovací úrovni) různé vlastnosti a různě detailní požadavky na vstup a výstup, zejména u heterogenního systému. Obecné parametry (identifikované na systémové úrovni) většinou nejsou vhodné pro posuzování regularity konkrétní vazby.
- Často se tedy stává, že parametry identifikované v 2. části úlohy vůbec při řešení této části nepoužijeme. (v popisech parametrů se použijí ty specifické pro danou vazbu, pro konkrétní prvky, které se s vazbou souvisí).

  PŘ: systém OKO, rozhraní: **čočka** -> **sítnice** (pokud to tak je)

  - **Čočka** je **vstupní** prvek vazby a **sítnice** je **výstupní** prvek vazby.
  - čočka propustí a zaostří paprsky světla a sítnice jej převede na obraz, z toho plynou 2 parametry
    - **parametr: "množství světla"** propuštěného (čočkou) versus množství světla potřebného k převodu na obraz (sítnice)
    - **parametr: "ostrost obrazu"** úroveň zaostření daná akomodací čočky versus požadavek na ostrý obraz na straně sítnice
  - Z výše uvedeného je jasné, že takové parametry se budou vyskytovat pouze na vazbě čočka -> sítnice a ne na ostatních vazbách (prvcích).
- parametry by měly mít **přímou souvislost** s identifikovanou vazbou. Nevadí když "přihodíte" parametry identifikované na vyšší úrovni (cena/váha/takt/barva/atp.), ale NESMÍ to být jediné parametry, se kterými budete pracovat. Najděte další specifické parametry, například pro systém POČÍTAČ, vazbu 1: *monitor-uživatel* by to mohly být tyto parametry:
  - příklad specifických parametrů pro vazbu 1: *rozlišení*, *jas*, *kontrast*
- **každý parametr charakterizujte**, popište jeho vstupní a výstupní podobu, například:
  - *rozlišení* .. fyzické rozlišení v pixelech které je monitor schopen zobrazit (případně interpolace nižších rozlišení), vzhledem rozměrům monitoru a (typické) vzdálenosti uživatele od monitoru vzniká požadavek na minimální rozlišení (obraz je hrubý, ale fonty a prvky na obraze jsou velké) i

na maximální rozlišení (obraz je jemný, ale fonty a prvky jsou na monitoru tak malé, že již rozlišení nevyhovuje, atp.)

- **u každého parametru zvolte, jakých hodnot nabývá** jak pro zdrojový prvek vazby, tak i pro cílový prvek vazby. Hodnoty parametru volte VŽDY pečlivě, MUSÍ to být konkrétní, může jít o:
    - hodnoty, např.: systém SPOJENÍ obsahující prvky spojovací matka-šroub; prvek: matka; parametr: průměr; hodnota: **6mm**
    - rozmezí hodnot, např. systém RÁDIO; prvek “ladící cívka”; parametr frekvence, hodnota: (rozsah) **88-108 MHz**
    - výčty kategorií, hodnot, rozmezí, např.: systém: DATOVÉ ÚLOŽIŠTĚ; prvek: databáze; parametr: formát dat; hodnota: (výčet) {**xml, json**}
- hodnoty parametrů na vstupu do prvku (IN, I) a výstupu z prvku (OUT, O)
    - POZOR! IN a OUT je o vstupu a výstupu prvků, ne o vstupu a výstupu vazeb, je to tedy zdánlivě obráceně (IN není vstup do vazby, ale výstup z vazby do prvku, tedy přesný opak).
    - **POZN**: IN a OUT je obecné, někdy je lepší místo těchto obecných označení použít konkrétní názvy prvků.



Obr. 3 ukázka vazby, s vyznačenými prvky OUT (zdrojový) a IN (cílový)

    - hodnoty parametrů na zdrojovém prvku vazby (OUT, O) odpovídají tomu, **co prvek do vazby “poskytuje”,** tedy jeho produkci (bez ohledu na to co je na druhé straně vazby očekáváno)
    - hodnoty parametrů na cílovém prvku vazby (IN, I) odpovídají tomu, **co prvek od vazby “očekává”,** tedy to co by chtěl dostat (když dostane něco jiného je problém = neregularita)

Výstupy této části

Pro OBĚ vazby platí stejná struktura. Struktura je opět dobrovolná, nezbytné je pouze dodržet očekávané výstupy.

3.1 Seznam zvolených vazeb

- tato část je společná pro obě vazby a obsahuje pouze výčet zvolených vazeb s volitelným krátkým komentářem.

  Např.: systém POČÍTAČ
    - vazba 1: *monitor* ***->*** *uživatel*
    - vazba 2: *základní deska* ***->*** *procesor*

- zvolené vazby by měly být **různého charakteru**, měly by na nich být rozpoznatelné (částečně) jiné parametry (nemusí být úplně všechny parametry jiné)

3.2 Vazba 1: monitor -> uživatel

[tato část návodu obsahuje veškerou další práci na jedné vazbě]

Parametry a hodnoty zvolených vazeb ve vstupní a výstupní matici

- tato část obsahuje výčet parametrů identifikovaných na vazbě, čím více a čím konkrétněji, tím lépe.
- parametry popište slovně (viz popis v kapitole “jak identifikovat parametry a jejich hodnoty”)
- parametry dále formálně popište dále pomocí **tabulky vstupních (O) a výstupních (I) parametrů vazby**.
    - hodnoty parametrů pro tabulky I a O mohou být stejné, pokud **v prvku nedochází k žádné transformaci** (prvek se na svém vstupu i na svém výstupu má stejné parametry se stejnými hodnotami). Příklad:
        - **PŘ:** systém VYSÍLAČKY, prvek: vysílačka, parametr: vysílací frekvence, hodnota: (výčet) 450 ÷ 451 MHz (například v kanálech po 20 kHz). Pokud budu komunikovat, lze předpokládat, že vstupní a výstupní frekvence na prvku bude stejná (na dané úrovni rozlišení)
        - **PŘ:** systém SPOJENÍ, prvek: šroub, parametr: průměr, hodnota: 6 mm. Nejedná-li se o nějaký mezikus, který má na jedné straně jiný průměr než na druhé, tedy o klasický šroub bude tento parametr stejný na vstupu i na výstupu z prvku šroub
        - potom není potřeba vytvářet 2 tabulky, ale vytvoří se pouze jedna, kde I=O
        - *opačným příkladem je například prvek “napěťový transformátor”, který na svém vstupu má napětí 230V a na svém výstupu napětí 12V.*
    - tabulka vstupních / výstupních parametrů obsahuje
        - jména prvků vazby (1. sloupec)
        - parametry a jejich hodnoty (v následujících sloupcích), pro tabulku I se jedná o parametry (resp. jejich hodnoty) požadované cílovými prvky vazby a pro tabulku O se jedná o parametry (resp. jejich hodnoty) generované zdrojovými prvky vazby
    - pokud se jedná o **jednosměrnou vazbu**, jsou důležité pouze hodnoty parametrů na výstupu (OUT) zdrojového prvku vazby a hodnoty parametrů na vstupu (IN) cílového prvku vazby. (viz následující příklad)
    - Protože popisujeme pouze JEDNU vazbu (relaci 2 prvků) bude tabulka parametrů obsahovat pouze DVA řádky, jeden za každý prvek vazby

Příklad tabulek pro systém POČÍTAČ a zvolenou vazbu *monitor-uživatel*

Tab. 5 Tabulka výstupních parametrů O, systém POČÍTAČ, vazba *monitor-uživatel*

O (**výstup z prvků**)	rozlišení	jas	kontrast	rychlost odezvy	barevné podání
uživatel	-	-	-	-	-
monitor	1024x768, 1200x900, 1440x1100, 1600x1200	30-250 cd/m^2	100-1000:1	5 µs	RGB

Tab. 4 Tabulka vstupních parametrů I, systém POČÍTAČ, vazba *monitor-uživatel*

I (**vstup do prvků**)	rozlišení	jas	kontrast	rychlost odezvy	barevné podání
uživatel	1600x1200	100 cd/m^2	400:1	2 µs	sRGB
monitor	-	-	-	-	-

POZN: Jak je vidět na proškrtnutých částech, jedná se o jednosměrnou vazbu, nemá smysl stanovovat parametry, resp. jejich hodnoty, pro VSTUP do monitoru (ten je na jiné vazbě, konkrétně vazbě *grafika-monitor*, a má jiné parametry) a pro VÝSTUP z uživatele (jedná se o koncový prvek této vazby)

Požadavky na regularitu vazeb, tzv. kritéria regularity

- každý parametr na vstupu a výstupu musí z hlediska regularity splňovat určité podmínky, jinak je vazba neregulární, tzv. kritéria regularity (KR)
- POZOR! Kritériem regularity nemusí být pouze rovnost.
- KR může být stanoveno jako (příklady):
    - hodnota parametru zdrojového prvku vazby (OUT) musí být **rovna** hodnotě parametru cílového prvku vazby (IN), př. šroub a matice.
      neboli: $val(název\ parametru)_{OUT} = val(název\ parametru)_{IN}$
      PŘ: $val(průměr)_{matice} = val(průměr)_{šroub}$
    - hodnota parametru zdrojového prvku vazby (OUT) musí být **menší nebo rovna** hodnotě parametru cílového prvku vazby (IN), př. výrobní linka.
      neboli: $val(název\ parametru)_{OUT} =< val(název\ parametru)_{IN}$
      PŘ: $val(kusů/h)_{montovna} =< val(kusů/h)_{lakovna}$
    - hodnota parametru zdrojového prvku vazby (OUT) musí být **větší nebo rovna** hodnotě parametru cílového prvku vazby (IN), př. zde, rozlišení.
      neboli: $val(název\ parametru)_{OUT} >= val(název\ parametru)_{IN}$
      PŘ: $val(rozlišení)_{monitor} >= val(rozlišení)_{uživatel}$

- všechny hodnoty parametru zdrojového prvku vazby (OUT) musí být **obsaženy** v hodnotách (výčtových) parametru cílového prvku vazby (IN), př. databáze

  neboli: $val(\text{název parametru})_{OUT} \in val(\text{název parametru})_{IN}$

  PŘ: $val(\text{formát})_{objednatel} \in val(\text{formát})_{dodavatel}$

- (alespoň jedna) hodnota parametru zdrojového prvku vazby (OUT) musí být **obsažena** v hodnotách (výčtových) parametru cílového prvku vazby (IN),

  neboli: $val(\text{název parametru})_{OUT} \cap val(\text{název parametru})_{IN}$

  PŘ: $val(\text{formát})_{objednatel} \cap val(\text{formát})_{dodavatel}$

Příklad: pro parametry a jejich hodnoty identifikované výše **definujme kritérium regularity.**

- rozlišení – alespoň takové, které chci, nebo právě takové? Záleží, co jako uživatel chci. Dejme tomu, že je požadavek 1600x1200 je požadavkem na minimální rozlišení.
  $val(\text{rozlišení})_{monitor} >= val(\text{rozlišení})_{uživatel}$
- jas – alespoň takové, které chci, nebo právě takové? Opět záleží, co jako uživatel chci. Dejme tomu, že je můj požadavek 100 cd/m^2 požadavkem na minimální jas.
  $val(jas)_{monitor} >= val(jas)_{uživatel}$
- jak jas, tak rozlišení by se daly vyjádřit i kritériem “musí obsahovat”, atp.

Matice anebo tabulka regularity

- jedná se o matici prvek-prvek, která obsahuje pouze analyzovaný výsek systému, v našem případě pouze 2 prvky (zdrojový a cílový prvek vazby)
- v případě regulární vazby tato matice obsahuje na místě této vazby hodnotu 0
- v případě neregulární vazby tato matice obsahuje na místě této vazby hodnotu 1 s přídavnou závorkou udávající ve kterých parametrech je vazba neregulární.

Tab. 5 Příklad matice neregularit

(matice R)	monitor	uživatel
monitor		1 (odezva, barva)
uživatel		

Odstranění neregularity

- tato část obsahuje zhodnocení nalezených neregularit a popis jejich odstranění pomocí: úpravy funkce prvku, vložením konverzního prvku, atp.
- pro každou vazbu popište více možností odstranění (zamyslete se nad všemi pěti teoretickými možnostmi odstranění regularity), a jakým způsobem by byly realizované

- v případě že je vazba regulární ve všech parametrech, tak tuto skutečnost konstatujte a vytvořte si neregularitu uměle (úvahou, co kdyby) a popište její odstranění. Rozhodně neuvádějte popis typu: “protože je můj systém funkční, tak mám všechno regulární, …”, chápu že to tak je, ale zadání zní, nalezněte neregularitu a odstraňte ji (i kdyby měla být uměle zavedená)

4 Dekompozice systému

Obecně

Na zvoleném systému proveďte tyto dekompozice: funkční, věcnou, topologickou a hierarchickou.

Jak na dekompozice

Při "dekompozici" tak, jak je pojata zde, tedy z již jednou vytvořeného systému (popsaného prvky a vazbami) provádíte POUZE seskupování prvků podle různých (funkčních, věcných, hierarchických, topologických) kritérií. Tento typ dekompozice bývá předstupněm integrační úlohy. Opačný pohled na dekompozici je ta, kterou provádíte při identifikaci systému, kdy postupujete od celku k částem podle zvoleného pohledu a rozlišovací úrovně.

Výstupem dekompozice je tedy organizační uspořádání prvků do skupin, **žádné prvky nemizí** ani se nepřidávají.

> *Aby při procesu dekompozice nedošlo k redukci systému, musí být dodrženy tyto postuláty:*
>
> - *postulát integrity ~ celek nesmí být zúžen, žádná část systému (např. významné vazby nebo funkce) nesmí být ztracena*
> - *postulát soudržnosti ~ musí být zachována možnost opětovného spojení, nesmí vzniknout izolovaná část.*
> - *postulát rovnoměrnosti, vyváženosti ~ mají vznikat srovnatelně složité podsystémy. (skripta SYSA, strana 74)*

Příklad: systém s 5ti prvky dekomponuji do 2 subsystémů A a B, z nichž A obsahuje 2 původní prvky a B obsahuje 3 původní prvky.

Funkční a věcná dekompozice

- věcná a funkční dekompozice lze provést vždy
- hledí pouze na prvky, jejich vazby nás nezajímají, to znamená, že do jednoho subsystému mohu dekomponovat prvky, které mezi sebou sice nemají vazbu, ale vykovávají podobnou funkci, či mají stejnou věcnou charakteristiku (barvu, tvar, čas, výplatu, …)
- **funkční** – hledám spojitost mezi vykonávanými funkcemi (v tabulce prvků) a obecnější funkcí, do které "spadají" konkrétní funkcionality více prvků systému.

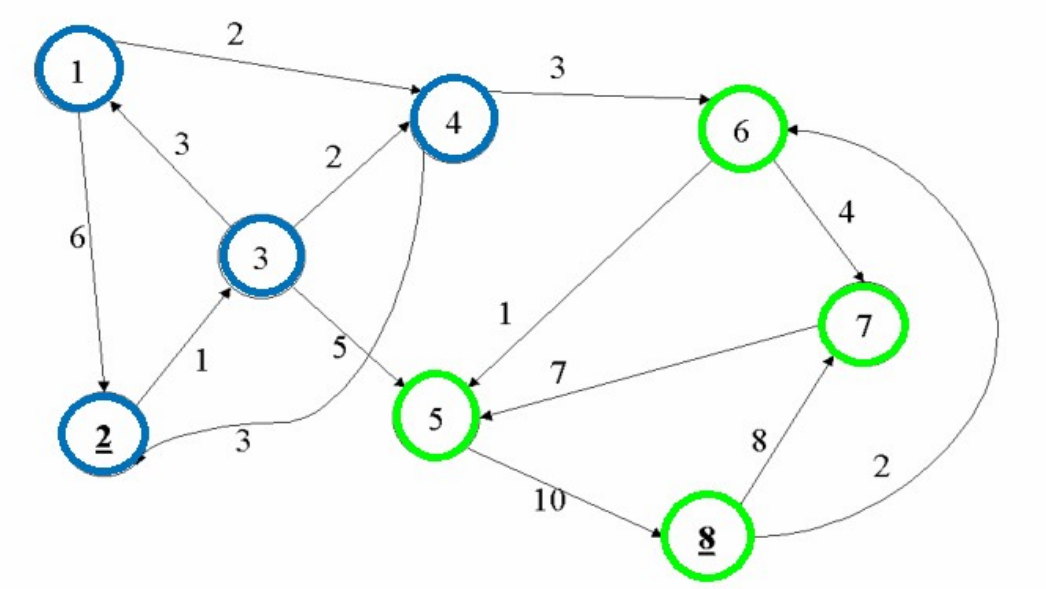
  PŘ: funkce prvků: 1:tisk, 2:zobrazení na monitor a 3:zvukový výstup => slučuji do funkce "výstup k uživateli"
- **věcná** … z tabulky prvků použiji jeden z identifikovaných věcných parametrů a podle něj prvky roztřídím

  PŘ: prvky s charakteristikou barva "roztřídím" do subsystémů podle jejich konkrétní barvy (systém o 8mi prvcích = růžová: 3 prvky, červená: 4 prvky a bílá: 1 prvek)

Topologická dekompozice

- nemusí být vždy proveditelná
- tato dekompozice hledá přirozené shluky v grafu systému, **vazby** jsou pro tuto dekompozici **nezbytné**, naopak, **funkce prvků** je pro tuto dekompozici **zcela zbytné.**
- postupuje se po vazbách a hledá tzv. minimální řez, tedy místo, kde existuje "slabá" vazba mezi hledanými subsystémy.
- Pravidlo pro nalezení subsystémů je:
    - nalezený "subsystém 1" má mezi svými prvky více vazeb než s ostatními prvky v systému
    - subsystém 1 a subsystém 2 mají přibližně stejně prvků
- při dekompozici vzniká problém se systémy typu: hvězda (střed a paprsky), kruh (propojené prvky) a všichni se všemi. Takovéto systémy většinou vůbec nejde topologicky dekomponovat, (existuje jen kruh, na všech paprscích hvězdy je pouze jeden prvek, atp.)

PŘ: Následující obrázek uvádí příklad topologické dekompozice, kdy na původním systému vznikají 2 subsystémy, jejichž prvky mají mezi sebou těsnější vazby než s okolím. Vazby 3 a 5 reprezentují minimální blokující řez. Vzniklé subsystémy zahrnují četné vazby mezi prvky subsystémů a reprezentují grafy, na kterých již topologická dekompozice nelze provést.



Obr. 3 ukázka topologické dekompozice systému na dva subsystémy (modrý a zelený)

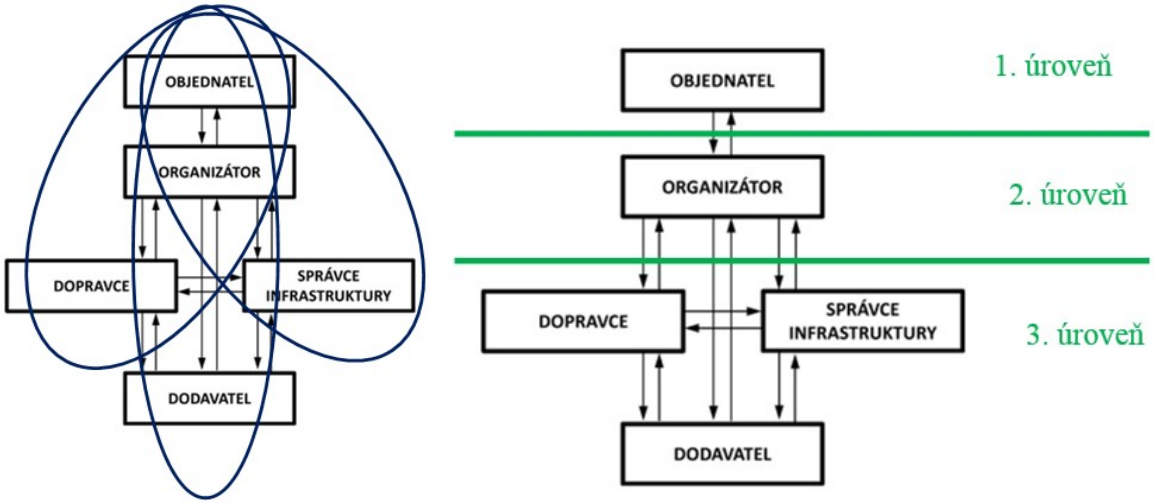
Hierarchická dekompozice

- nemusí být vždy proveditelná
- aby byla tato dekompozice proveditelná, musí Váš systém vyhovovat zároveň dvěma podmínkám:
    1. Je to strukturní systém, grafem struktury je strom A
    2. Významné relace tohoto systému vyjadřují nadřízenost / podřízenost.

- Identifikace hierarchického systému na reálném objektu povahy fyzikální, chemické či technické (s výjimkou systémů AI) **je zpravidla nevhodná**, zatímco v úrovni ekonomické, sociální, právní, politické a někdy i biologické může být vhodná.
- **hledají se vztahy nadřízenosti a podřízenosti**, které mohou, ale nemusí existovat, dekomponuje se:
    a. do úrovní nebo
    b. do větví.

U většiny systému si lze nějakou hierarchii představit, například plochou (všechny prvky jsou si rovny či mají jen jeden nadřízený prvek) nebo košatou (hierarchie státních firem).

PŘ: hierarchický systém objednávání veřejné dopravy skládající se z pěti prvků lze dekomponovat do 3 hierarchických úrovní či 3 větví.



Obr. 4 ukázka hierarchické dekompozice systému do úrovní (vpravo) či do větví (vlevo)

PŘ: následující příklady jsou pokusy o hierarchickou dekompozici funkčně identifikovaných systémů. U funkčních systémů není hierarchická dekompozice vhodná, přesto lze, pokud je to nezbytně nutné, provést (doporučujeme ji u těchto systémů nedělat).

- funkční systém park. automat dekomponuji do úrovní (1) řídicí jednotka / (2) to ostatní.
- funkční systém hra na kytaru dekomponuji do úrovní: (1) kytarista / (2) kytara, zesilovač
- funkční systém kytara dekomponuji do úrovní (1) tělo kytary / (2) struny, napínací mechanismus strun, zesilovač, senzor

Výstupy z této části

- Ke **každé** dekompozici vytvořte **výčet VŠECH** dekomponovaných prvků rozdělených do “subsystémů”
- dekompozice **znázorněte graficky**, například obarvením prvků systému a popisem v legendě, zejména důležité u hierarchické a topologické dekompozice.
- pokud nelze dekompozice provést, to se týká zejména topologické či hierarchické, stručně okomentujte, proč nelze provést. (aby nechyběla ve výčtu dekompozic)

5 Chování systému a genetický kód

Obecně

Na systému identifikujte a popište procesy charakterizující chování systému.

Postup:

- Na systému nalezněte vstupní vazby do systému (vnější události, události přicházející z okolí systému).
- Pro každý z těchto vstupů nalezněte všechny procesy, které jsou danou událostí aktivovány. Pro zjednodušení neuvažujte paralelní procesy (v určitém okamžiku může být aktivovaná jenom jedna vnější událost).
- Pro každou událost i sestrojte přechodové podgrafy a vypište dílčí stavové prostory S_i.
- Pro každý dílčí stavový prostor S_i sestrojte matici dílčího chování systému D_i a vypište množiny dílčího chování F_i (množina všech procesů (cest) mezi vstupními a výstupními prvky, $F_i=\{f_{1i}; f_{2i}; ... f_{ni}\}$).

Na systému identifikujte genetický kód systému.

Postup:

- Vypište úplný stavový prostor S ($S = S_1 \cup S_2 \cup S_3$ ...) a
- celkové chování systému (formou matice chování systému a množiny chování systému).
- Nalezněte genetický kód systému jako průnik dílčích chování. Pokud nelze tímto způsobem určit, najděte silné funkce systému a genetický kód nalezněte jako množinu procesů s nejvyšším počtem silných funkcí.

Komentář ke zpracování

- vstupní události aktivují systém prostřednictvím jeho vazby s okolím, události tedy přichází z konkrétního, identifikovaného okolí.
- vstupních událostí bývá celá řada. Vám stačí nalézt a popsat alespoň 3, které vedou na rozdílné chování v systému, je tedy možné, že budou přicházet z různých okolních prvků systému.
- při volbě událostí postupujte od těch nejpravděpodobnějších k těm méně pravděpodobným.
- nevybírejte události vedoucí na vzájemně, alespoň částečně, exkluzivní chování.
- chování jsou procesy vyvolané vstupní událostí skládající se z postupně aktivovaných stavů v systému. **Stav systému může:**
    - **odpovídat aktivaci jednoho prvku** = *na prvku P1 se něco děje a je tedy aktivován, následně je aktivován prvek P2 atp*.
    - **odpovídat aktivaci více prvků zároveň** = *prvek P1 a P2 jsou aktivovány současně (tvoří tedy JEDEN stav) a poté je aktivován prvek P2 samostatně a poté třeba prvky P1 a P3 současně. Stavy SE NEROVNAJÍ prvkům.*

    - **odpovídat specifické akci daného prvku** = *nejprve prvek P1 sečte, pak prvek P1 vydělí, následně prvek P2 vynásobí atp. Aktivace jednoho prvku, podle toho co se na něm děje vede k různým stavům!!!*
- to, jaký zvolíte způsob popisu stavů (viz odrážka výše), je na Vás, ale tento způsob pak musíte dodržet v průběhu celého řešení.
- mezi prvky, které zastupují po sobě aktivované stavy v chování, by v systému měla existovat vazba. Jak by měl například aktivovaný prvek A aktivovat další prvek (například C) pokud mezi nimi v systému neexistuje vazba?
- stavy v chování musí být identifikovatelné (ztotožnitelné) s prvky systému, nevolte takové stavy, které nemají ve vašem systému ekvivalent v prvku!
- **DŮLEŽITÉ**: volte vstupní události a následná chování tak, abyste každý prvek v systému aktivovali (byl součástí nějakého stavu) alespoň v jednom chování.
- popis chování musí obsahovat informaci o tom, co se v daném chování mezi prvky identifikovanými v systému děje.
- přechodový graf musí dostatečně popisně identifikovat stavy, které reprezentuje
- stavový prostor je pouze výčet stavů v daném chování
- matice dílčího chování je čtvercová matice reprezentující přechody mezi stavy, v řádcích a sloupcích budou tedy stavy (ne prvky).
- dílčí chování systému je pouze jiný zápis přechodového grafu, přepis z vertikální do horizontální podoby
- stavový prostor celého systému - pozor na duplicitu stavů při sloučení všech dílčích stavových prostorů
- matice chování celého systému SD – opět pozor na případnou duplicitu stavů ve sloupcích / řádcích matice

Návod k nalezení Genetického kódu

- nejprve hledáme průnik procesů (nebo částí procesů) přes všechny procesy dílčích chování F_i
    - najdeme je průnikem standardních matic chování SDi avšak tím zjistíme jenom průniky procesů délky 2
    - proto pak ještě ověříme, zda nalezené části procesů na sebe nenavazují
- pokud je výsledek prázdná množina:
    - možný postup: snížíme náročnost podmínky – místo ve 100% dílčích chování hledáme v méně procentech a procenta postupně snižujeme, až nalezneme ucelený proces. Doporučení: není dobré jít pod 50%
    - další možnost - použít fuzzy přístup (např. zohledním podobnosti různých posuzovaných stavů, kdy prohlásíme některé stavy za podobné)
- POZN.: v některých případech nelze genetický kód takto nalézt (např. proto, že v rámci domácí úlohy byl vybrán příliš malý počet a zcela různorodých chování) Pokud nelze tímto způsobem určit, najděte silné funkce systému (při jejich volbě zohledněte smysl popisovaného systému, jeho hlavní resp. primární funkce a

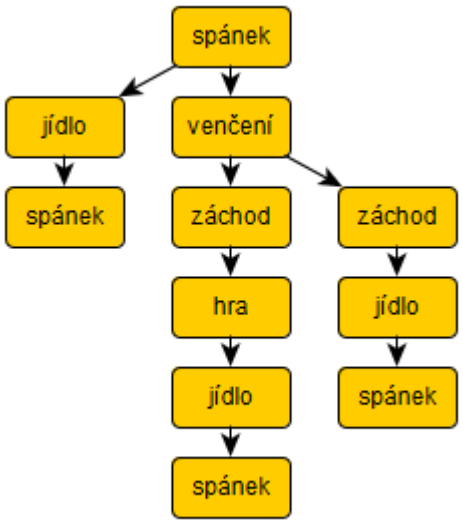
také frekvenci jejich aktivací) a genetický kód nalezněte jako množinu procesů s nejvyšším počtem silných funkcí.

PŘ (značně zjednodušené jedno chování):

POZN.: Zde je strukturní systém poskládán z prvků vykonávajících jednotlivé funkce související se stavem psa, tedy: **spánek, jídlo, atd**. **NE**: pes, člověk, vodítko, miska s krmením atp. Nicméně dokázali bychom si představit stavy chování spánek, venčení, jídlo, atd. související s prvky systému nepřímo, kdy by prvky systému mohly být například zahrada (na venčení), předsíň (na spánek), kuchyň (na jídlo), lesík (na záchod), atd. **Klíčové je dokázat identifikovat prvky systému se stavy v chování a mít mezi těmito prvky vazby** (tam kde jsou vazby v chování).

Událost: člověk (mimo systém) budí psa a chce jej nakrmit.

Chování: člověk budí psa ze spánku a podle jeho momentální nálady (psa) s ním jde ven jej vyvenčit (kde pes vykoná potřebu) a případně si s ním zahrát, následně mu dává najíst a ukládá k odpočinku, případně s ním nejde ven a dává mu jen najíst.



Přechodový graf: (vpravo) ->

Dílčí stavový prostor:

S_1 = {spánek, venčení, záchod, hra, jídlo}

Dílčí chování systému zapsané množinou procesů dílčího chování:

F_1: f_{11}= spánek->venčení->záchod->hra->jídlo->spánek

f_{12}= spánek->venčení->záchod->jídlo->spánek

f_{13}= spánek->jídlo->spánek

Dílčí chování systému zapsané maticí dílčího chování: viz 6

Tab. 6 Matice dílčího chování

D_1	Spánek	venčení	záchod	hra	jídlo
spánek	-	1			1
venčení		-	1		
záchod			-	1	1
hra				-	1
jídlo	1				-

Výstupy z této části

- alespoň **3 vstupní události** identifikované názvem vedoucí na různá chování
- pro každou vstupní událost analyzujte následné chování systému a formálně zapište:
    - nestrukturovaný slovní popis daného chování pomocí procesů aktivovaných touto událostí, v rozsahu cca 1 odstavec textu,
    - přechodový graf procesů aktivovaných danou událostí,
    - dílčí stavový prostor
    - matici dílčího chování
    - dílčí chování systému F_i formou množiny procesů
- stavový prostor celého systému S (vektor)
- matice chování celého systému SD (matice)
- množina chování systému F (formální zápis)
- postup jeho nalezení genetického kódu
    - slovní popis nalezení viz výše
- genetický kód zapsaný:
    - formou matice SD_i a označenou G
    - vypsáním všech nalezených procesů tvořících genetický kód
- krátký slovní závěr úlohy
    - zda výsledek dává smysl, nebo ne
    - pokud ne, kde nastala chyba či zjednodušení, apod.)