

## 1 Metody strukturované analýzy a návrhu systémů

Východiskem metod systémové analýzy byly a jsou metody návrhu a realizace programových systémů a je tedy přirozené, že tyto metody souvisejí s principy algoritmických programovacích jazyků. Metody strukturované analýzy a návrhu jsou založeny na principech procedurálních programovacích jazyků. Díky dlouholetému vývoji a zkušenostem s praktickými aplikacemi se soubor metod strukturované analýzy a návrhu ustálil na několika základních, s jejichž využitím lze popsat velmi široké spektrum systémů.

Metody strukturované systémové analýzy a návrhu představují ne příliš rozsáhlý soubor vzájemně se doplňujících metod založených na relativně jednoduchých a jasně specifikovaných principech. Důsledné uplatňování takovýchto principů sice neposkytuje uživateli na první pohled příliš mnoho volnosti, avšak vede ke tvorbě konzistentních modelů. Metody strukturované systémové analýzy jsou plně v souladu s principy obecné teorie systémů [Žampa95]. V neposlední řadě jejich nespornou výhodou je existence ověřených a propracovaných prostředků počítačové podpory, které tyto metody implementují. Ačkoliv s příchodem objektově orientovaných metod na počátku 90. let se další vývoj metod strukturované systémové analýzy a návrhu více méně zastavil, díky všem výše zmíněných aspektům si dodnes uchovávají přední pozici v oblasti analýzy a návrhu jak softwarových, tak především nesoftwarových systémů.

S ohledem na v této práci užitý CASE systém case/4/0<sup>1</sup> bude přehled metod strukturované analýzy a návrhu systémů z větší části vycházet z jejich implementace v tomto programovém prostředí [c40Met]. Tvůrci systému case/4/0 vycházeli z metodiky publikované Edwardem Yourdonem [Yourdon89] označované jako SA/SD – Structured Analysis/Structured Design a z myšlenek, poznatků a prací dalších známých osobností, jako například Michaela Jacksona, Toma DeMarca, Paula Warda, Steva Mellora, ....

### 1.1 Metody strukturované analýzy systémů

Jedním z principů uplatňovaných pro zvládnutí složitosti reálných systémů je primární popis kvalitativních vlastností systému a následné zachycení jeho vlastností kvantitativních, tedy princip odděleného a postupného studia struktury a chování systému. Analýza obecných systémů je, až na výjimky tvořené primitivními systémy, v úvodní fázi vždy analýzou jejich struktury, na kterou následně navazuje specifikace chování.

Existuje však třída aplikací, mezi něž patří například modelování systémů managementu jakosti, u nichž je požadováno studium systémů pouze na úrovni kvalitativních vlastností vedoucí ke tvorbě strukturních modelů. Vzhledem k zaměření této práce bude hlavní pozornost v popisu metod systémové analýzy věnována metodám popisu struktury systémů a metody popisu chování budou zmíněny pouze přehledově.

#### 1.1.1 Funkční struktura

Metoda analýzy funkční struktury je jednou ze základních metod strukturované analýzy používaná především k popisu struktury systému. Jedná se o metodu grafickou, která slouží k zachycení hierarchické dekompozice systému na subsystémy a prvky pomocí stromových

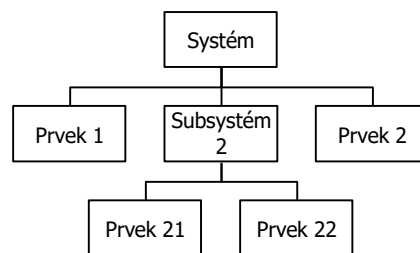
---

<sup>1</sup> CASE systém case/4/0 je produktem německé společnosti microTOOL (<http://www.microtool.de>). Je prostředkem počítačové podpory analýzy a návrhu založené na strukturovaném přístupu.

diagramů. Hierarchické stromové uspořádání vyjádřené vzájemnými vazbami prvků zachycuje vztah nadřazenosti a podřízenosti prvků a vztah sounáležitosti podřízených prvků patřících k jednomu prvku nadřazenému. Listové položky hierarchického stromového diagramu představují elementární prvky systému, tedy prvky s ohledem na daný účel tvorby modelu dále nedekomponované. Pro plně popsanou strukturu systému tvořenou souborem listových položek je možno s využitím principů grafického zápisu algoritmů podle Jacksona v jisté míře popsat jejich algoritmický obsah.

Diagram funkční struktury obsahuje zdánlivě velmi málo informace o analyzovaném systému, neboť zachycuje pouze hierarchickou nadřazenost a podřízenost prvků a postihuje pouze jejich základní souvislosti. Zcela mimo oblast zájmu této metody stojí vzájemné informační vazby jednotlivých prvků systému.

Jednoduchost a přehlednost takto získaného popisu funkční struktury systému vytváří z této metody vhodný nástroj pro první, orientační představu o analyzovaném systému. Ve srovnání s dále zmíněnou metodou informačních toků na menší ploše zachycuje a zobrazuje větší část analyzovaného systému. Metoda analýzy funkční struktury může být využita samostatně, častěji však jako vhodný doplněk metody informačních toků.



Funkční struktura

### 1.1.2 Informační toky

Metoda informačních toků je základní metoda strukturované analýzy. Tato metoda je plně v souladu s poznatky obecné teorie systémů [Žampa95], dokonale vyhovuje obecné definici systému a důslednému vymezení kvalitativních a kvantitativních vlastností systému. Jedná se o grafickou metodu, která formou hierarchicky uspořádaných síťových diagramů vyjadřuje dekompozici systému na subsystemy a prvky a současně dovoluje zachytit informační vazby mezi těmito prvky. Jedná se proto o metodu, pomocí které lze studovat především strukturální vlastnosti systému.

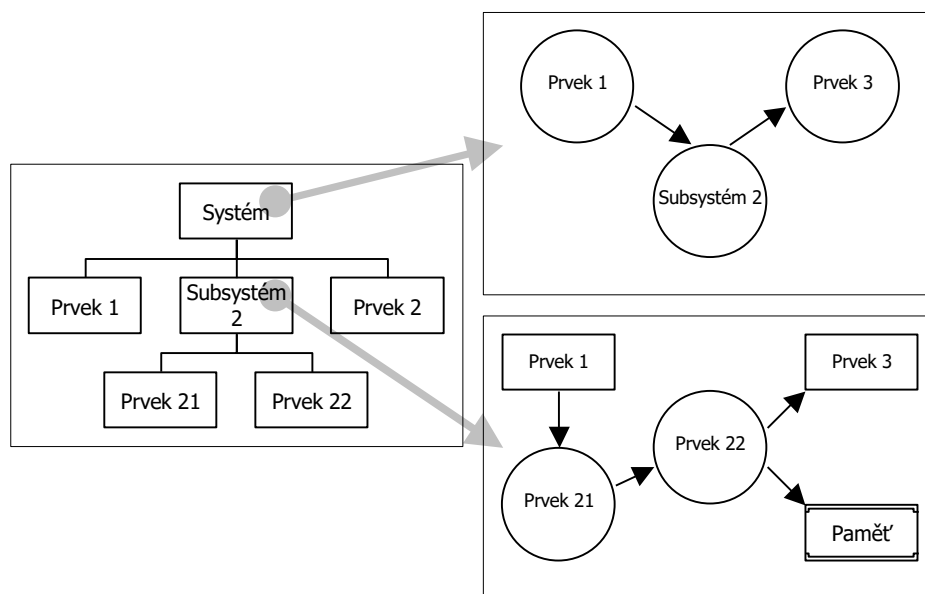
Metoda informačních toků je metodou výlučně grafickou. Pro zvýšení přehlednosti a tím i vypovídací hodnoty diagramu informačních toků je účelné rozlišit prvky systému podle typu. V základním členění lze prvky rozdělit na prvky aktivní a pasivní.

Základními aktivními prvky jsou funkční prvky neboli funkce, prvky zajišťující transformaci vstupní informace na výstupní. Aktivní prvky lze dále rozlišovat na prvky příslušné k popisovanému systému a prvky, které lze považovat vzhledem k popisovanému systému za vnější. Tyto vnější prvky jsou součástí definice vazeb popisovaného systému na okolní svět. Jejich vnitřní aktivita není předmětem studia systému, jejich funkce je dána a není úkolem návrhu systému je vytvářet nebo měnit. Představují adekvátní model okolí popisovaného systému a jsou vzhledem k základnímu předpokladu obecné teorie systémů na uzavřenost systému [Žampa95] jeho nezbytnou součástí. Není nutné, ale může být účelné, od základních

funkčních prvků odlišit prvky systému, které sice náležejí k vnitřním prvkům systému, ale které nejsou bezprostřední součástí právě studované části a patří ostatním částem systému.

Pasivní prvky představují paměti. Jedná se o prvky, které jsou schopny uchovat uloženou informaci. V případě softwarově orientovaných systémů mohou být realizovány například soubory nebo databázemi, v oblasti nesoftwarových systémů například protokoly, seznamy nebo záznamovými knihami.

Významnou složkou diagramu informačních toků jsou informační vazby mezi prvky systému – informační toky. Obsah informačního toku nemůže být s ohledem na požadavek přehlednosti diagramu vyjádřen zcela detailně. To však není na závadu, neboť složitosti a principu postupného poznávání systému vyhovuje i hierarchicky organizovaný způsob postupného zpřesňování popisu předávané informace. Informační tok je vyjádřen spojnicí příslušných prvků systému doplněnou o komentář, který charakterizuje předávanou informaci způsobem, který odpovídá aktuální úrovni hierarchické dekompozice systému. Detailní popis předávané informace je většinou záležitostí popisu datových struktur nebo datového modelu.



Informační toky

Metoda informačních toků vyjadřuje formou hierarchicky uspořádaných síťových diagramů dekompozici systému na subsystémy a prvky a současně zachycuje informační vazby mezi těmito prvky. Na vrcholu této hierarchie stojí tzv. kontextový diagram, který vyjadřuje začlenění systému do souvislostí okolního světa. V popisu metody analýzy funkční struktury systému byla zmíněna případná vazba na metodu informačních toků, jsou-li obě metody v rámci analýzy uplatňovány. Jednotlivé diagramy informačních toků jsou vázány k prvkům – uzlům hierarchické stromové funkční struktury a podrobněji popisují jednu úroveň jedné větve.

Metoda informačních toků v systému case/4/0 užívá notace podle DeMarca a je rozšířena o základní principy Ward-Mellorovy metody. DeMarcova metoda je sama o sobě efektivně využitelná pro jednodušší systémy, neboť umožňuje pouze jednoúrovňové zobrazení prvků systému a jejich vazeb. Ward-Mellorova metoda rozšiřuje standardní metodu informačních toků o řídicí funkce a řídicí toky a doplňuje ji tak o možnost hrubého popisu chování systému

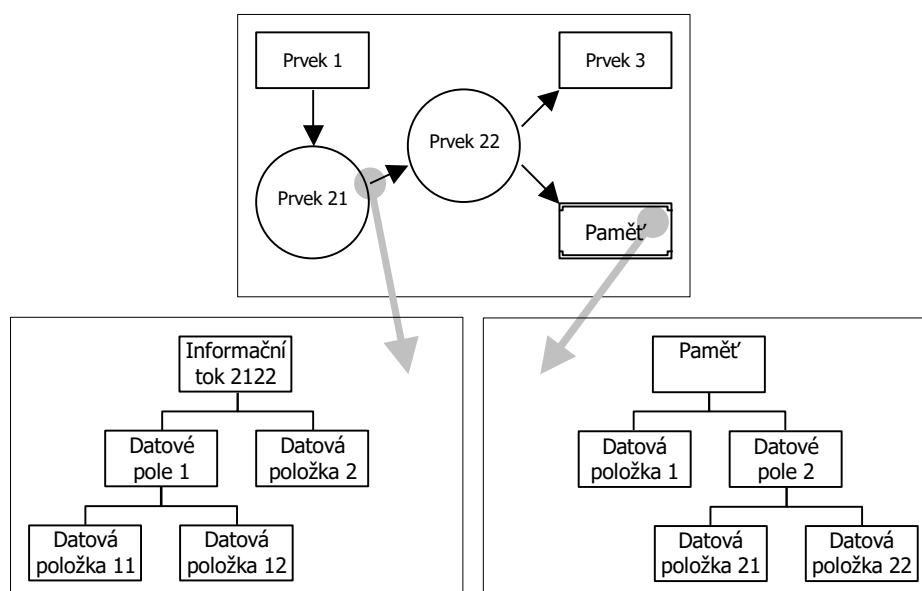
spočívajícího v zachycení sekvence vykonání ucelených a relativně samostatných činností. Je-li to z hlediska pochopení funkčnosti systému nezbytné, lze k řídicím funkcím připojit stavový diagram. Tento přístup však s sebou přináší nutnost násilného oddělování informačních a řídicích toků již od prvních kroků analýzy. Vyjádření řídicích toků je zřetelným přechodem od popisu struktury systému k popisu jeho chování a dochází tedy k prolínání popisu kvalitativních a kvantitativních vlastností systému.

I přes výše uvedené možnosti je metoda informačních toků určena především k zachycení strukturálních vlastností systému, popis chování je ve většině případů omezen nebo potlačen. V aplikacích vedoucích ke tvorbě strukturálních modelů je vhodným a dostačujícím popisem chování systému výstižné pojmenování funkčních prvků nebo slovní popis ve formě komentáře k příslušným prvkům.

### 1.1.3 Datové struktury

Metoda analýzy funkční struktury a metoda informačních toků neposkytují vhodné a dostatečné prostředky pro analýzu datové složky systému. Jak vazby mezi prvky systému popsané informačními toky, tak i pasivní prvky systému mohou představovat složité a rozsáhlé datové struktury.

Jedním z prostředků datové analýzy je popis datových struktur. Analýza datových struktur umožňuje pomocí hierarchických stromových diagramů postupné rozčlenění informačních toků nebo paměťových prvků. Všechny prvky tohoto diagramu představují obecně chápané bloky informace – datové položky. Na analýzu struktury dat bezprostředně navazuje detailní datová analýza, která je prostředkem k podrobnému popisu elementárních datových položek – listových prvků hierarchického stromového diagramu. Detailní datová analýza umožňuje přiřadit každé elementární položce datové struktury datový element, který určuje formu uložení informace. Pro datové elementy je možno specifikovat například typ, rozsah nebo výčet možných hodnot.



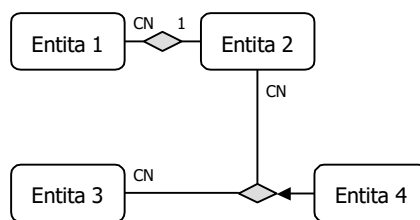
Datové struktury

Diagramy datových struktur stejně jako diagramy funkčních struktur využívají v systému case/4/0 hierarchických stromových diagramů s využitím zobecněných principů grafického zápisu algoritmů podle Jacksona, které umožňují vytvářet složitější datové struktury při zachování přehlednosti, například datové struktury s podmíněnými datovými položkami.

#### 1.1.4 ER model

ER model – model entit a jejich vzájemných vztahů je speciální, avšak velmi rozšířenou metodou datové analýzy. Je vhodný pro analýzu systému v případě, kdy složitost systému spočívá spíše ve složitosti struktury dat než ve složitosti jeho funkčních složek. Nachází uplatnění ve specializované aplikační oblasti nazývané hromadné zpracování dat.

ER model zachycuje formou síťového grafu objekty reálného světa a vztahy mezi nimi. Množiny objektů reálného světa mající shodné vlastnosti se nazývají entitami, vztahy mezi nimi pak relacemi. Entity mohou být blíže specifikovány množinou atributů, které mají shodný význam jako datové elementy užívané při detailní datové analýze. Relace mezi entitami jsou specifikovány kardinalitou a těsností vazby. ER modely se využívají pro tvorbu modelů dat na logické neboli konceptuální úrovni, tedy modelů dat nezávislých na jejich fyzické realizaci prostřednictvím specifického databázového systému.



ER model

ER modely by neměly být jediným prostředkem analýzy systémů, ačkoliv tomu tak u aplikací z oblasti hromadného zpracování dat většinou bývá. Východiskem by měla být vždy analýza systému metodou informačních toků, ve které bude specifikován pasivní paměťový prvek, který bude následně detailně analyzován a popsán pomocí ER modelu. ER model použitý jako jediný prostředek analýzy systému je účelovým zjednodušením pohledu na systém a může vést k opomenutí významných požadavků řešené úlohy.

#### 1.1.5 Metody popisu chování

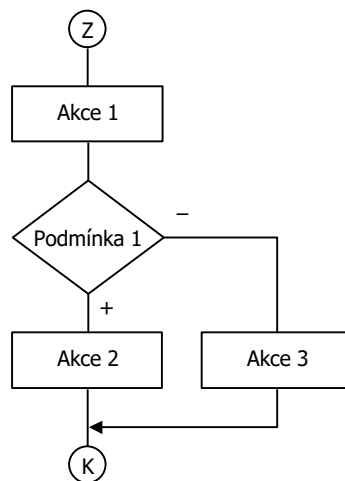
Popis chování systému je v obecném slova smyslu jeho algoritmizací. Z tohoto faktu vychází názor, že popis chování systému, zejména pak softwarově orientovaného, je předmětem jeho návrhu a obsahem vlastního programového řešení. Algoritmus chování je však v určitých případech nutné nebo účelné podrobněji popsat již ve fázi analýzy. Metody popisu chování umožňují ve fázi analýzy zachytit algoritmickou složku systému, avšak abstrahují od konkrétního způsobu realizace, který je předmětem fáze návrhu.

Metody popisu chování jsou obvykle velmi příbuzné metodám užívaným při tvorbě programů, neboť, jak bylo zmíněno, jde v obou případech o popis algoritmů. Základním prostředkem by mohl být tedy některý z vyšších programovacích jazyků. Obvykle se však používají formalizované jazyky stojící mezi běžným jazykem a jazykem programovacím,

kteře jistou měrou omezují mnohotvárné vyjadřovací schopnosti přirozeného jazyka, avšak neváží úzce na syntaxi určitého programovacího jazyka.

Další metody popisu chování vycházejí z obecně platného poznatku efektivnějšího zpracování graficky kódované informace. Metody jsou založeny na grafickém vyjádření nebo na vhodné kombinaci grafického vyjádření a formalizovaného textového popisu. Obvykle je pomocí grafických prostředků zachycena základní struktura algoritmu, která je pro specifikaci algoritmu na detailní úrovni doplněna relativně krátkými sekvencemi příkazů zapsanými formalizovaným jazykem.

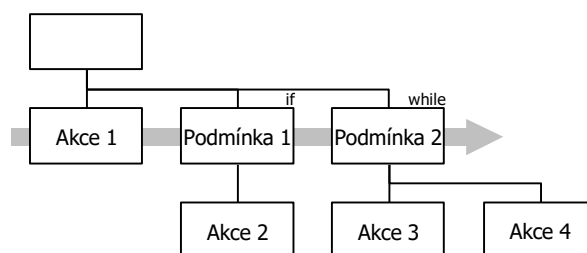
Klasickým grafickým prostředkem zápisu algoritmu je vývojový diagram. Díky své jednoduchosti získal oblibu v nejrůznějších oblastech, které daly vzniknout jeho různým modifikacím. Nejjednodušší varianta zachycuje základní strukturu algoritmu pomocí vzájemně propojených elementů představovaných bloky a podmínkami. Detailní specifikace algoritmu je obsahem příslušných elementů a je vyjádřena formalizovaným jazykem.



Vývojový diagram

Při nedodržení určitých pravidel mohou konstrukce zachycené pomocí vývojových diagramů odporovat zásadám strukturovaného programování, a tím mohou znesnadňovat případnou následnou programovou realizaci.

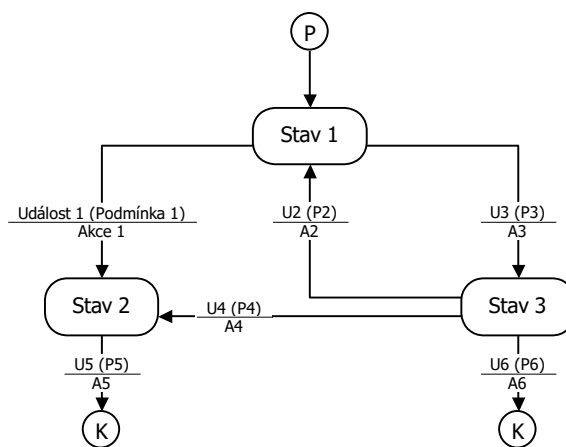
Mezi metody, které podporují a dodržují zásady strukturovaného programování patří metoda grafického zápisu algoritmů podle Jacksona, označovaná jako Jacksonovy diagramy. Základní struktura algoritmu je popsána hierarchickým stromovým diagramem, detailní specifikace je obsahem příslušných elementů a je vyjádřena formalizovaným jazykem.



## Jacksonův diagram

O metodě grafického zápisu algoritmů podle Jacksona bylo pojednáno v souvislosti s popisem metody analýzy funkční struktury systému. Pro listové položky diagramu funkční struktury bylo možno s využitím této metody specifikovat jejich algoritmický obsah.

Metoda stavových diagramů je grafická metoda popisu chování použitelná pro vyjádření jednodušších algoritmů nebo pro popis algoritmů na hrubé úrovni. Systém je popsán síťovým grafem, jehož uzly znázorňují stav systému a hrany naznačují možné přechody mezi stavy s vyjádřením podmínek přechodu a akcí s přechodem spojených. Akce a stavy systému pouze symbolicky označují funkce a jejich účinky. Algoritmickou náplň akcí je však nutné popsat s využitím jiných prostředků.



Stavový diagram

Metoda stavových diagramů byla zmíněna při popisu metody informačních toků a její implementace v programovém systému case/4/0 jako nástroj detailní specifikace řídicích funkcí.

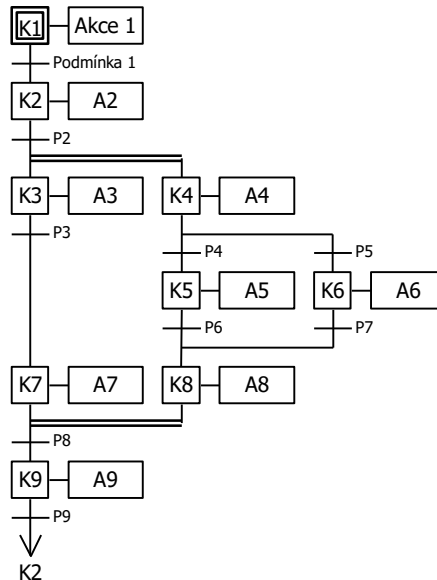
Metoda sekvenčních funkčních grafů<sup>2</sup> představuje grafickou metodu popisu chování systému vyhovující nejobecnějším nárokům na popis algoritmu. Základy metody jsou postaveny na principech Petriho sítí. Původně byla metoda vyvinuta pro detailní zachycení RT vlastností systémů, tedy pro popis algoritmů, které kladou velké nároky na vzájemnou i časovou synchronizaci. Se značným časovým zpožděním se tato metoda dostala do povědomí i jako jedna z analytických metod pro popis chování systému.

Metoda sekvenčních funkčních grafů využívá k popisu systému obdobně jako metoda stavových diagramů síťový graf. Zásadní rozdíl však spočívá v tom, že uzlem síťového grafu není stav systému, ale krok algoritmu, respektive akce s daným krokem spojená. Přechod z kroku do kroku je vázán podmínkou přechodu a aktivitou kroků této podmínce bezprostředně předcházejících. Síťový graf tedy zachycuje základní strukturu algoritmu, detailní popis je pak obsahem akcí spojených s dílčími kroky. Ačkoliv detailní algoritmický

---

<sup>2</sup> Metoda funkčních kroků a přechodů byla poprvé uveřejněna v roce 1977 pod názvem GRAFCET (Grphe Fonctionnel de Commande Etape-Transition). V současné době se však s touto metodou lze setkat častěji pod názvem SFC (Sequential Function Chart).

popis není povinnou součástí analýzy metodou sekvenčních funkčních grafů a lze v případě potřeby zůstat na úrovni kvalitativního popisu, není tato metoda vhodným nástrojem pro vlastní analýzu systému, respektive pro její počáteční fázi. Velmi dobře však může navazovat na metodu informačních toků nebo na popis hierarchické funkční struktury.



Sekvenční funkční graf

Vzhledem k tomu, že metoda sekvenčních funkčních grafů byla a je využívána jako grafický programovací jazyk, je zřejmé, že přechod od analýzy k detailnímu návrhu bude přímý a tedy vysoce efektivní.

## 1.2 Metody strukturovaného návrhu systémů

Metody strukturovaného návrhu systémů nacházejí uplatnění především v oblasti softwarově orientovaných systémů. Vzhledem k zaměření této práce budou metody strukturovaného návrhu zmíněny pouze přehledově.

Z hlediska své podstaty lze odlišit metody na podporu návrhu datové a algoritmické části systému. V programovém systému case/4/0 jsou pro návrh datové části systému k dispozici typové struktury a relační model dat, pro návrh algoritmické části pak modulové struktury spolu s implementačními stromy.

Typové struktury představují technickou analogii datových struktur zmíněných v části analytických metod. Primárně jsou odvozeny od datových struktur a popisují strukturu dat užívaných funkcemi v modulových strukturách. Pro popis typových struktur jsou využity stejné grafické prostředky jako pro popis datových struktur, tedy hierarchické stromové diagramy.

Zatímco ER modely jsou ve fázi analýzy užívány pro tvorbu modelů dat na logické neboli konceptuální úrovni, relační model dat je nástrojem návrhu databází zohledňujícím aspekty fyzické realizace prostřednictvím specifického databázového systému. Relační model shodně jako ER model využívá pro zobrazení modelu dat formu síťového diagramu.



Modularita je jedním z obecných principů redukce složitosti a v případě softwarových systémů umožňuje dosáhnout jejich přehlednosti, srozumitelnosti a udržitelnosti. Modulová struktura se používá pro popis softwarových jednotek – modulů. V grafické formě pomocí hierarchického stromového diagramu umožňuje zachytit obsah modulů, tedy funkce jako algoritmické jednotky, data, reference na moduly a elementy uživatelského rozhraní. V programovém systému case/4/0 lze jednotlivým elementům modulové struktury přiřadit generovací funkce, které zabezpečí převod informace zachycené v modulové struktuře do podoby zdrojového kódu v požadovaném programovacím jazyce. Součástí systému case/4/0 jsou generovací funkce například pro programovací jazyky COBOL, C nebo BASIC. Díky otevřenosti tohoto programového systému je však uživateli dána možnost jejich modifikace nebo tvorby vlastních.