

AoM33PIS - Průmyslové informační systémy

Přednáška č. 6

30. 3. 2016



Katedra Kybernetiky K13133

Centrum znalostního managementu K13393

Agenda

- Modelování podnikové architektury.
- Analýza a modelování informačních toků ve výrobním podniku.
- Metodologie SSADM, IDEF.
- Enterprise Engineering, DEMO.

Modelování podniku

- **SYSTÉM** je množina vzájemně souvisejících objektů či komponent, na kterou nahlížíme jako na celek a která byla vytvořena pro určitý účel; systém je ohraničený, a to co je vně této hranice se nazývá okolí, s nímž systém interaguje prostřednictvím svých rozhraní.
- Základní úlohou analýzy systému je nacházení faktů, zákonitostí a omezení v systému. Prvním úkolem při analýze je stanovení hranice systému. Softwarový produkt je pak správním či řídicím systémem nad systémem základním (pozor na směřování pojmů!)

Metody modelování podniku

- shora dolů (**top-down**)
založeno na **specializaci**: začíná na úrovni komplexního systému, který postupně dekomponuje na subsystemy, které jsou dále dekomponovány atd. Analytický proces je zobrazitelný **stromovou strukturou** s originálním systémem v kořeni.
- zdola nahoru (**bottom-up**)
založeno na **generalizaci**: začíná identifikací a analýzou několika jednoduchých struktur, snaží se najít jejich **společné charakteristiky** a chování a **zobecnit** je tak, aby původní byly jejich speciálními případy.

Analýza při modelování podniku

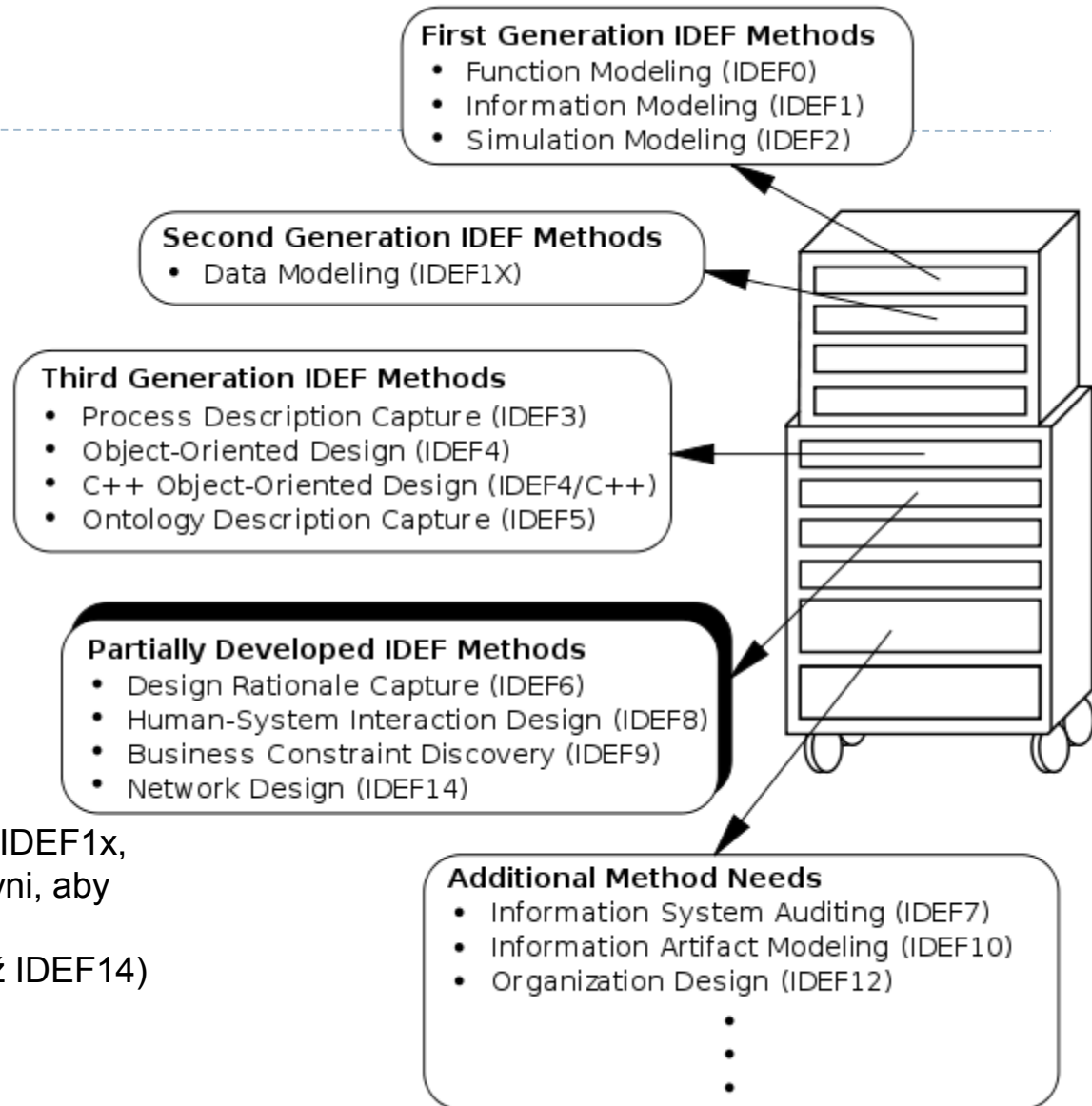
- Obvyklým předmětem analýzy je
 - a) existující systém („AS-IS“), jehož struktura a funkce nejsou zřejmé zákazníkovi a řešiteli, případně zákazník neumí strukturu a chování systému řešiteli vysvětlit,
 - b) neexistující systém („TO-BE“), o němž má zákazník nepříliš přesnou představu a neumí požadované funkce řešiteli vysvětlit.
- Inženýrská praxe řeší v naprosté většině případů realizaci takto formulovaných systémů.
- Produktem analýzy je specifikační dokumentace (*Projektová dokumentace, Funkční specifikace, Vnější specifikace, Specifikace návrhu, Dokument požadavků* aj.)
- Specifikační dokumentace je **závazným podkladem** pro návrh a realizaci systému. Často má **právní platnost** (je přílohou obchodní smlouvy jako doklad, který přesně definuje předmět a rozsah plnění).

Metody modelování podniku

- Metody, které používají funkční dekompozici, podporují identifikaci komponent v systému, které jsou více zaměřené na funkce (stanovené cíle).
- OO přístup z principu podporuje systémové komponenty více kompatibilní s datovými modely.
- Funkcionálně zaměřený analytik bude identifikovat jiné moduly, než OO analytik.
- Velké množství metod, metodik, metodologií, modelovacích jazyků
- **Dva světy – strukturovaná analýza versus objektově orientovaná analýza**
- Současné metody strukturované analýzy mají potenciál konvergovat směrem k OO – viz např. Object Oriented SSADM.
- Metodologie může zahrnovat metody funkční (strukturované) analýzy i OO – viz např. IDEF.
- Pro modelování podnikových procesů existuje řada metodik, např. **IDEF3**, případně specializované metodiky a systémy pro **BPM – např. ARIS.**

IDEF

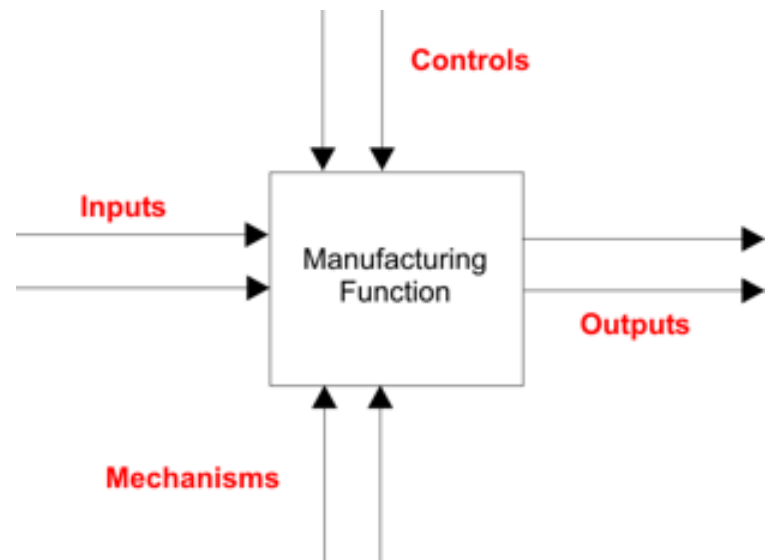
- ▶ **IDEF – The Integrated DEFinition** metodologie je souborem metod, jejíž účelem je *komplexně podporovat potřeby podniků v oblasti modelování podnikové architektury*.
- ▶ Metodika je produktem *výzkumného programu ICAM* (Integrated Computer-Aided Manufacturing program), který byl původně vytvořen US AirForce za účelem *zlepšení koordinace výrobních operací* (1970/80).
- ▶ V současné době je v rámci IDEF rozvinuto šest metod (IDEF0, IDEF1, IDEF1x, IDEF3, IDEF4, IDEF5) na takové úrovni, aby mohly být použity v praxi. Dalšíh osm dalších metod (IDEF6 až IDEF14) je pak ve vývoji.
- ▶ příklady implementace:
 - ▶ CA ERwin Process Modeler – podpora IDEF0, IDEF3
 - ▶ CA ERwin Data Modeler – podpora IDEF1X
 - ▶ Sybase PowerDesigner – podpora IDEF1X



IDEF	Metodika	Poznámka (stav v r.2009)
IDEF0	Function modeling	plně vyvinuto a používáno od r.1995
IDEF1	Information Modeling	plně vyvinuto a používáno od r.1995
IDEF1X	Data Modeling	plně vyvinuto a používáno od r.1995
IDEF2	Simulation Model Design	poslední změny v r.1995
IDEF3	Process Description Capture	plně vyvinuto a používáno od r.1995
IDEF4	Object-Oriented Design	plně vyvinuto a používáno od r.1995
IDEF5	Ontology Description Capture	plně vyvinuto a používáno od r.1995
IDEF6	Design Rationale Capture	poslední změny v r.1995
IDEF7	Information System Auditing	nezměněno oproti inic.verzi
IDEF8	User Interface Modeling	poslední změny v r.1995
IDEF9	Business Constraint Discovery	poslední změny v r.1995
IDEF10	Implementation Architecture Modeling	nezměněno oproti inic.verzi
IDEF11	Information Artifact Modeling	nezměněno oproti inic.verzi
IDEF12	Organization Modeling	nezměněno oproti inic.verzi
IDEF13	Three Schema Mapping Design	nezměněno oproti inic.verzi
IDEF14	Network Design	poslední změny v r.1995

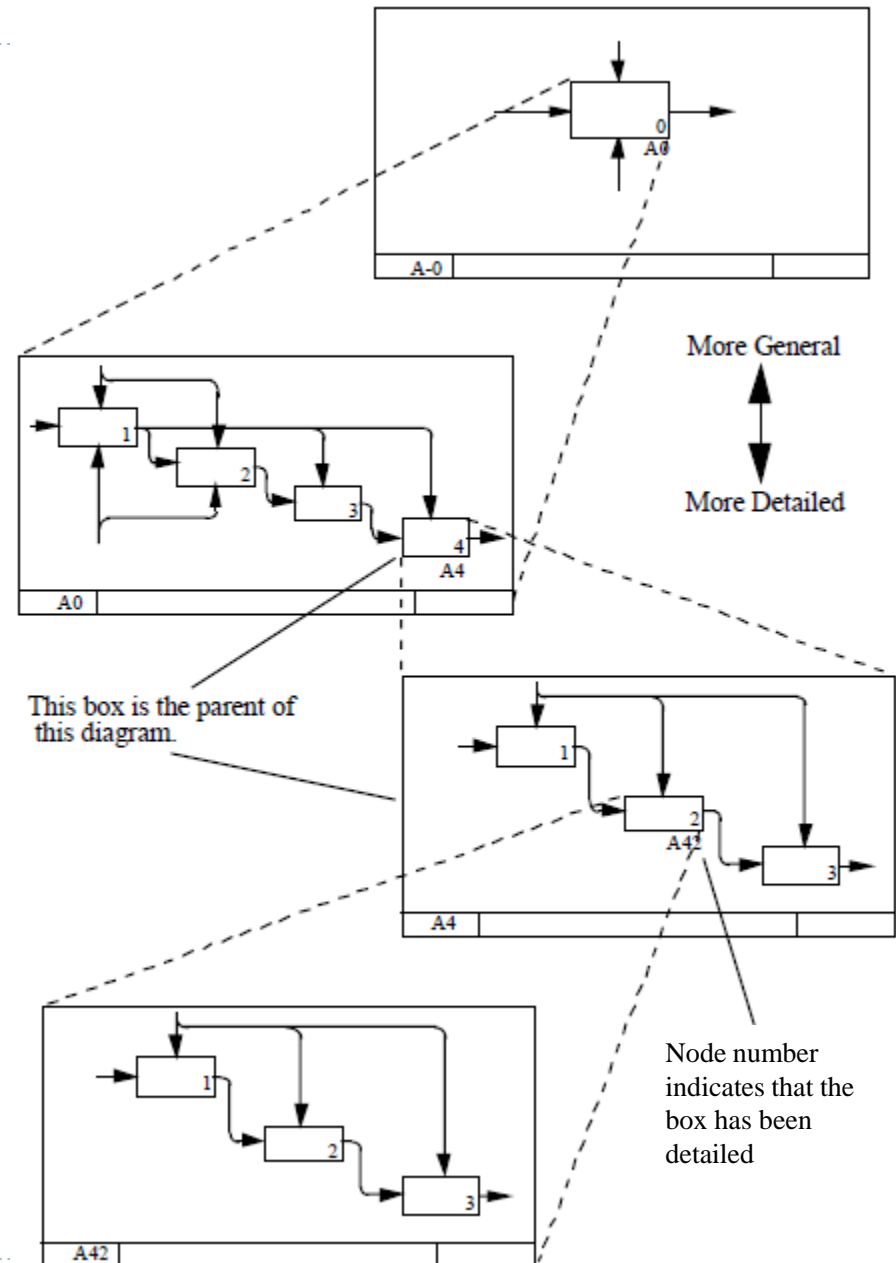
IDEFO

- ▶ IDEFO - používá se ke specifikaci funkčních modelů podniku, které slouží k popisu základních činností podniku.
- ▶ V modelech se znázorňují hlavní *aktivity* a jejich *vstupy*, *výstupy*, *řídící vstupy* a *mechanismus* spojený s každou hlavní aktivitou → základní stavební komponenta funkčních modelů **ICOM** (Input, Control, Output, Mechanism)
 - ▶ vstupy (Inputs) – data nebo objekty k transformaci funkcí
 - ▶ výstupy (Outputs) - funkcí transformovaná data nebo objekty
 - ▶ řízení (Controls) – podmínky vyžadované pro provedení transformací
 - ▶ mechanismus (Mechanism) – prostředky a způsoby užití k provedení transformací



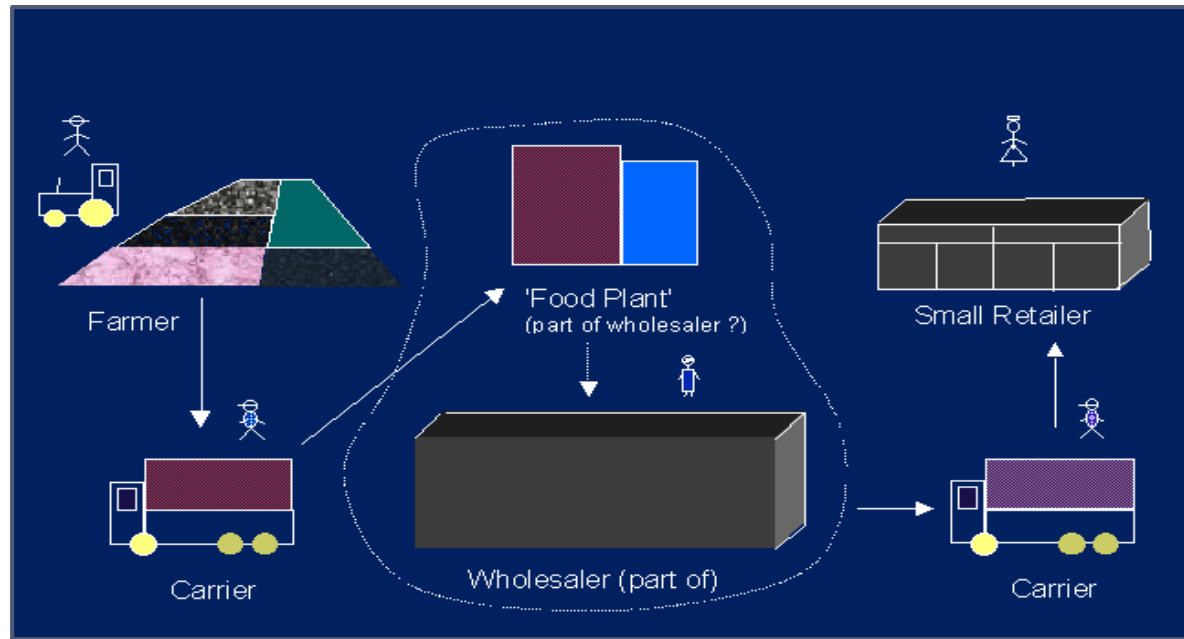
IDEFO

- ▶ Hierarchická podstata IDEF0 - usnadňuje schopnost stavět (AS-IS) modely, které mají reprezentaci a interpretace shora-dolů, ale jsou založeny zdola-nahoru na analýze procesů.
- ▶ Standardní postup analýzy existujícího podniku (AS-IS):
 - ▶ interview s odborníky v podniku, v problémové doméně
 - ▶ sdružování aktivit, které jsou úzce spojené nebo funkčně podobné
 - ▶ tvorba hierarchie na základě aktivit, sdružených do skupin a úrovní, až po konceptuální úroveň
- ▶ Standardní postup analýzy nově vytvářené podnikové architektury (TO-BE)
 - ▶ konstrukce postupuje shora-dolů
 - ▶ začínáme na konceptuální úrovni,
 - ▶ postupujeme logickou dekompozicí, rekurzivně až na požadovanou úroveň detailu



V I T E

The Fruit & Vegie Virtual Enterprise



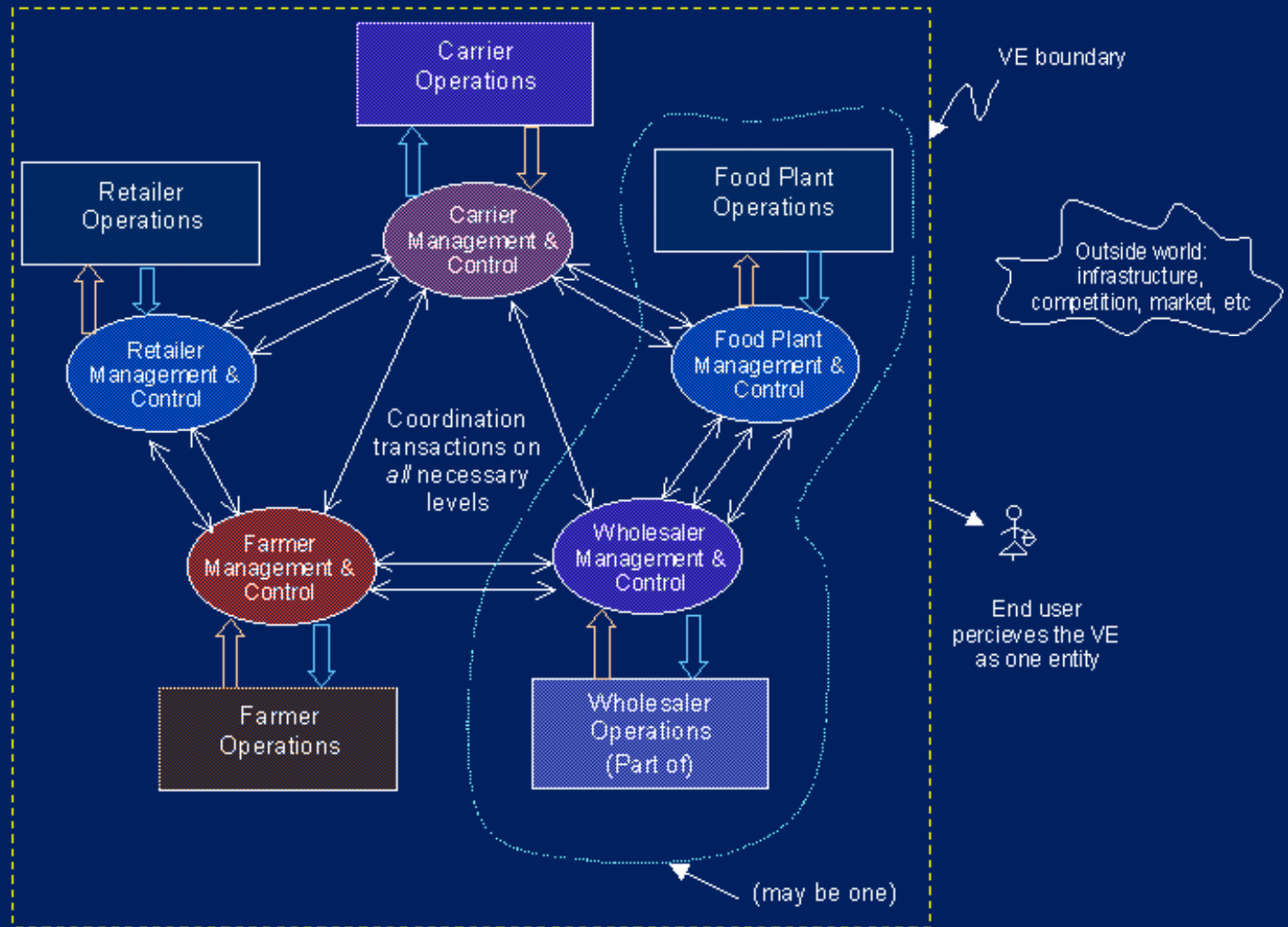
- ▶ integrovaný dodavatelsko-odběratelský řetězec, ve

kterém *farmář (farmer)*, *dopravce (carrier)*, *zpracovatel (food-plant)*, *velkoobchod (wholesaler)* a *prodejce (retailer)* spojují zdroje a strategie za účelem výroby a prodeje produkce;

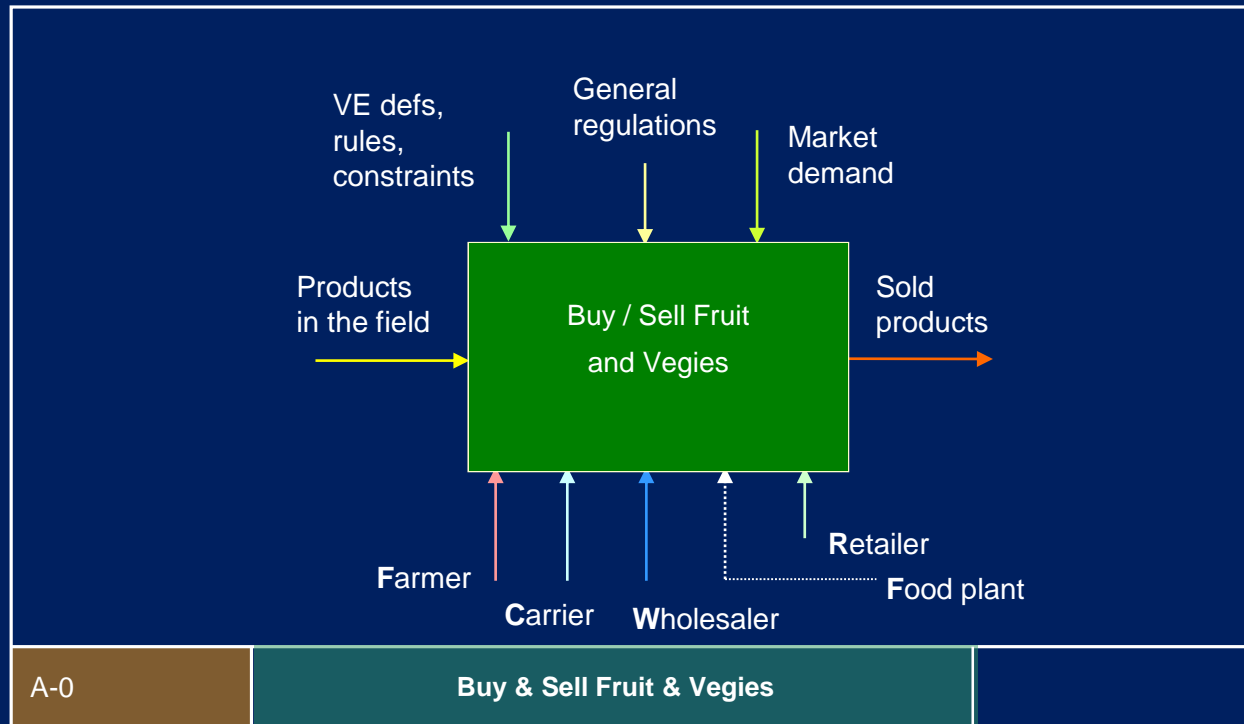
- ▶ v tomto příkladu, zpracovatel realizuje skladování, třídění a balení produkce; mohl by být samostatným subjektem nebo součástí velkoobchodu.

IDEFO - příklad

Virtuální organizace funguje jako autonomní celek pro účely svého produktu

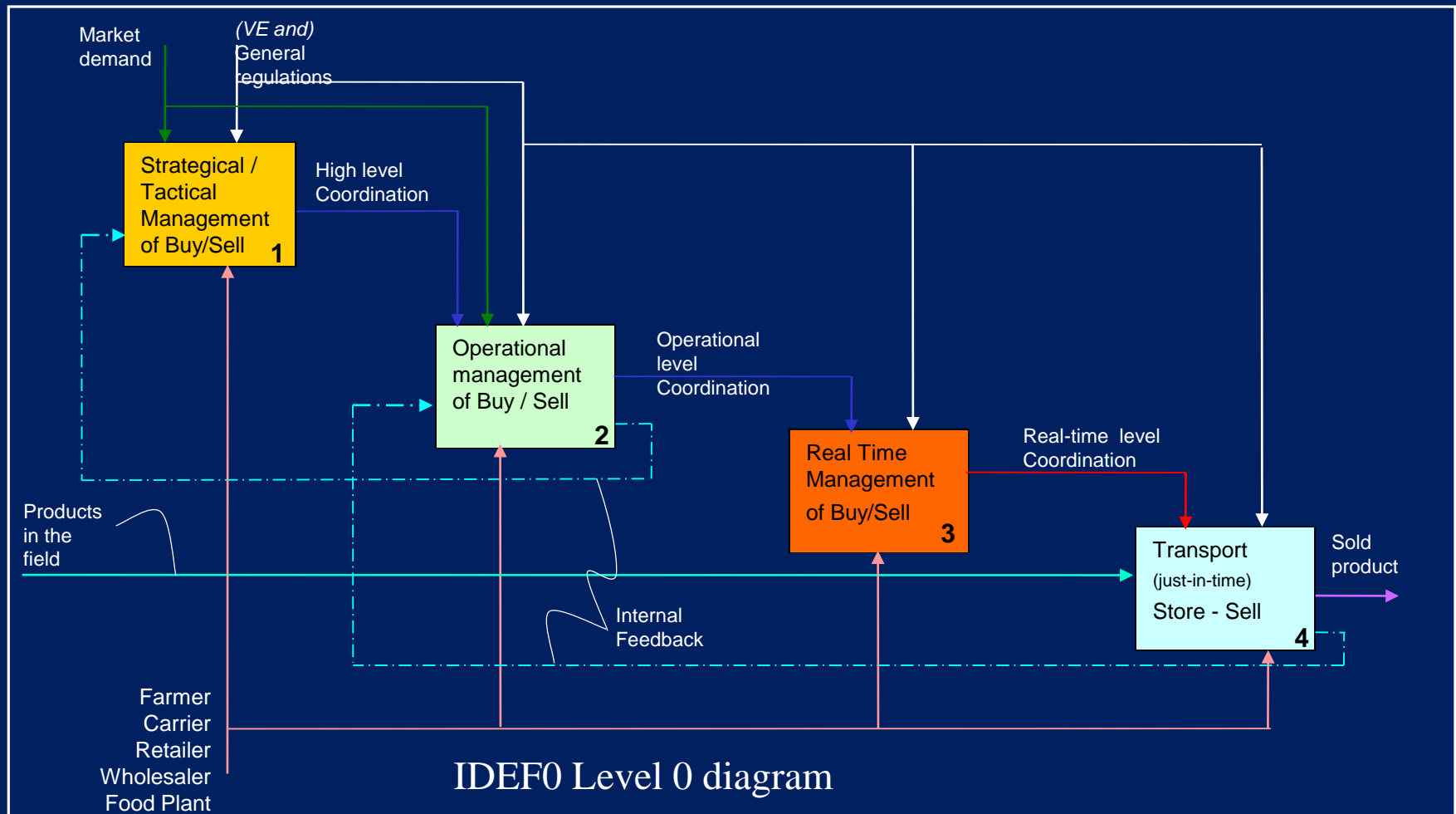


IDEFO - příklad



IDEFO Level A-0 (context) diagram

IDEFO - příklad

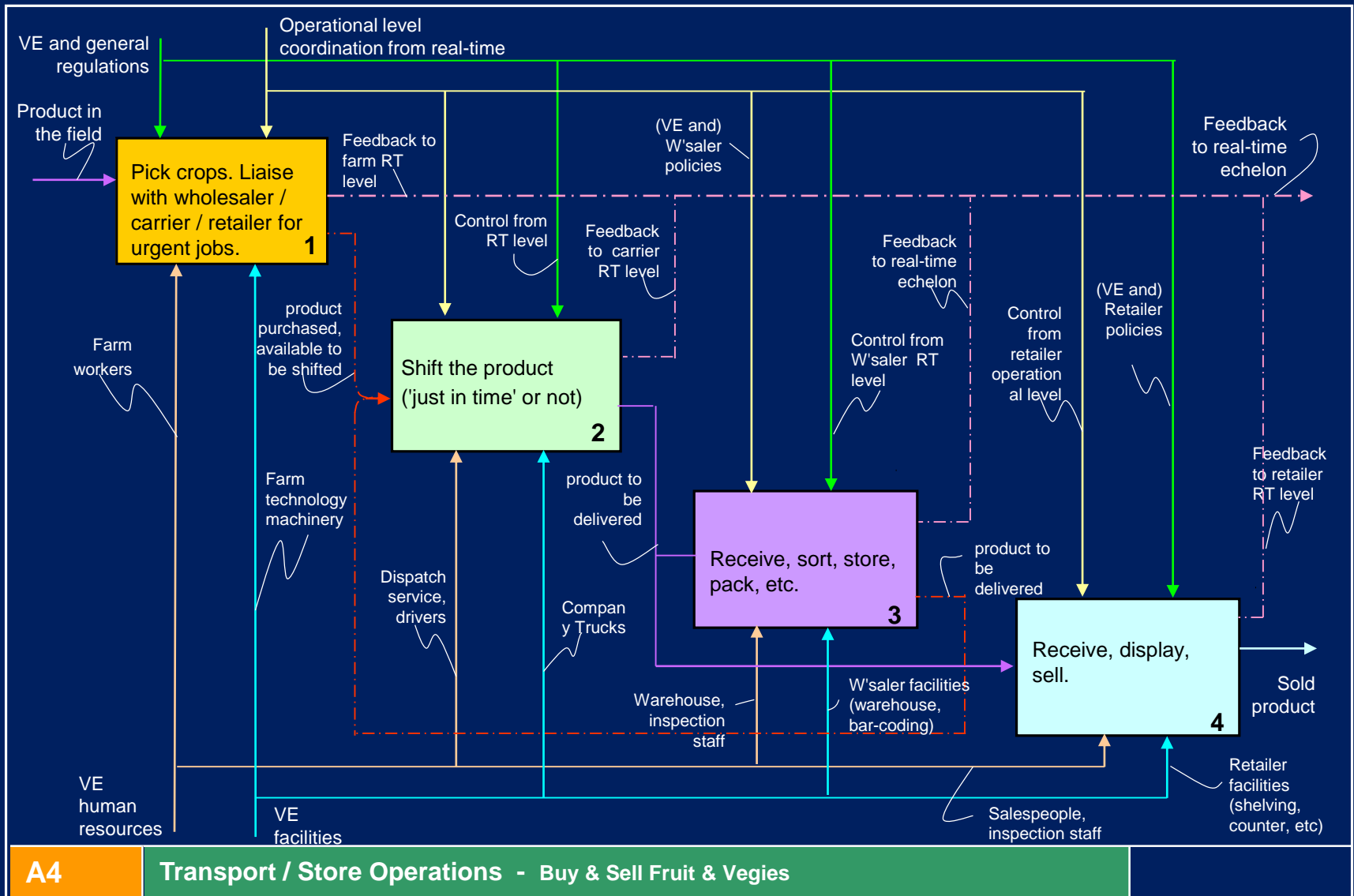
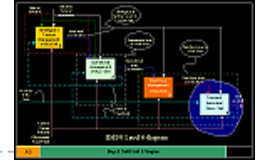


IDEFO Level 0 diagram

A0

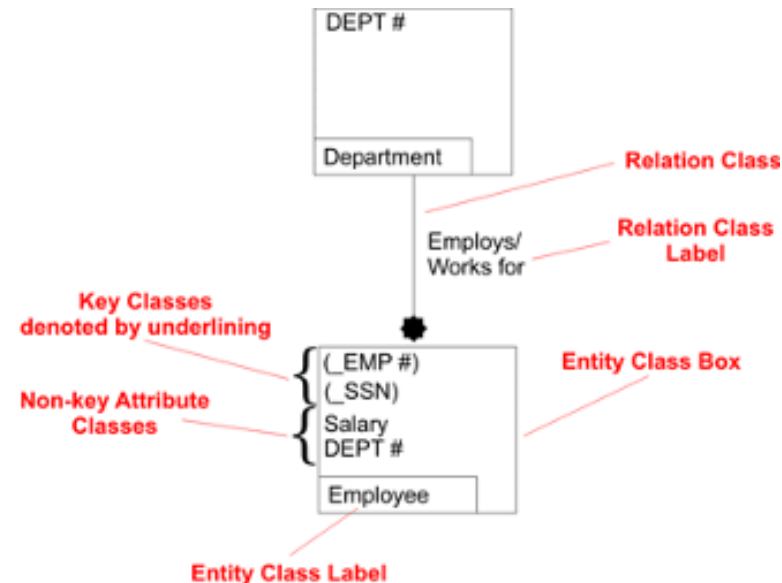
Buy & Sell Fruit & Vegies

IDEFO - příklad



IDEF1

- ▶ **IDEF1** - slouží pro **modelování informací** v podniku, zejména ve fázi **formulace požadavků**. Modely identifikují pojmy, které jsou v podniku používány a vztahy mezi nimi (např. pojem materiál, pojem sklad materiálu a vztah materiál se nachází ve skladu materiálu).
- ▶ IDEF1 **není metodou pro navrhování datových modelů**, ale slouží organizaci k identifikaci, porozumění a správě informačních zdrojů.
- ▶ Postup při analýze požadavků musí rozlišit:
 - ▶ **Reálný svět**, vnímaný lidmi v organizaci, složený z fyzických a konceptuálních objektů (např. lidé, místa, věci, nápady, atd.), vlastnosti těchto objektů a vztahy souvisejících s těmito objekty.
 - ▶ **Informační obraz**, zachycující informační stopu objektů, nacházejících se v reálném světě, tj. sbírané, ukládané a spravované informace. IDEF1 je omezen na popis tohoto obrazu.
- ▶ IDEF1 se používá pro:
 1. identifikaci, **jaké informace** jsou aktuálně spravované v organizaci,
 2. identifikaci, **které z problémů** objevených při analýze požadavků jsou způsobeny nedostatky ve správě relevantních informací,
 3. specifikaci, které informace mají být **spravovány** v případě **"TO-BE"** modelování



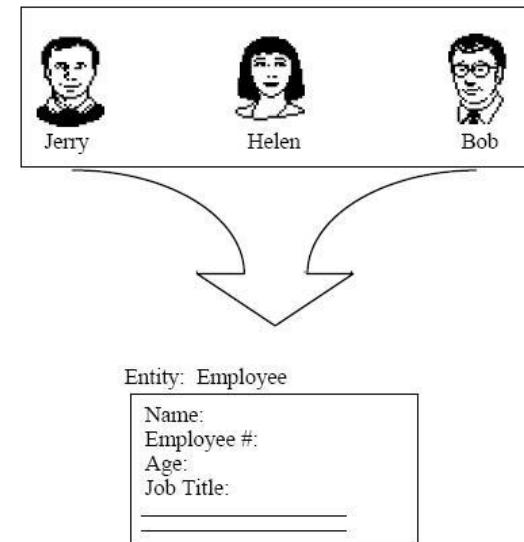
IDEF1X

- ▶ IDEF1X – určeno pro návrh relačních datových modelů.
- ▶ Využívá tři pohledy:
 - ▶ Externí schema pro uživatelské pohledy ;
 - ▶ Konceptuální schema, integrující externí schemata ;
 - ▶ Interní schema definující fyzické struktury.

▶ Fáze tvorby modelu:

- ▶ Fáze 0 – iniciace projektu
 - ▶ definice projektu (projektový plán), zdrojových materiálů, konvencí pro modelování
- ▶ Fáze I – definice entit
 - ▶ identifikace entit v problémové doméně

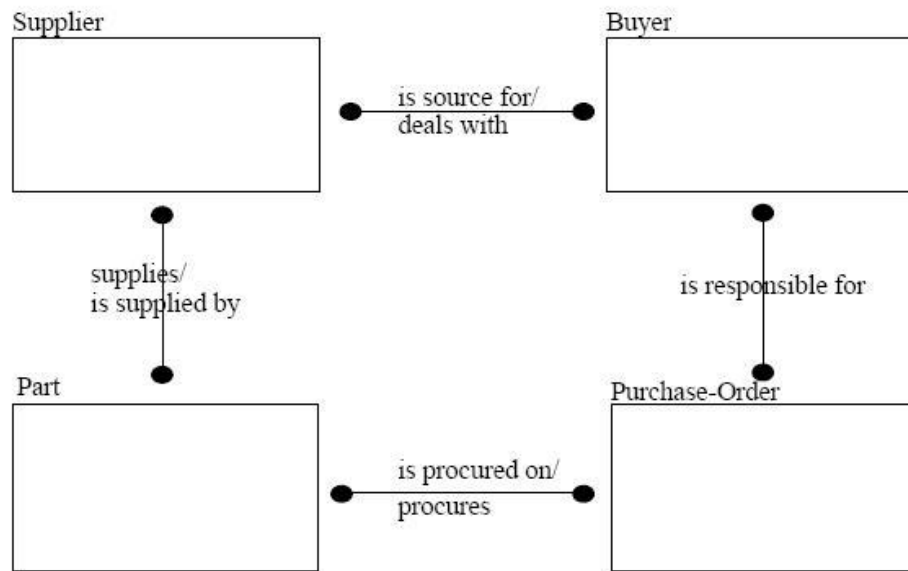
Entity Instances



IDEF1X

- ▶ Fáze II – definice relací
 - ▶ identifikace základních vztahů mezi entitami (upřesňováno v dalších etapách)
 - ▶ primární výstupy:
 - Relationship definitions, Entity-level diagrams, Relationship matrix

Entity Level Diagram Example



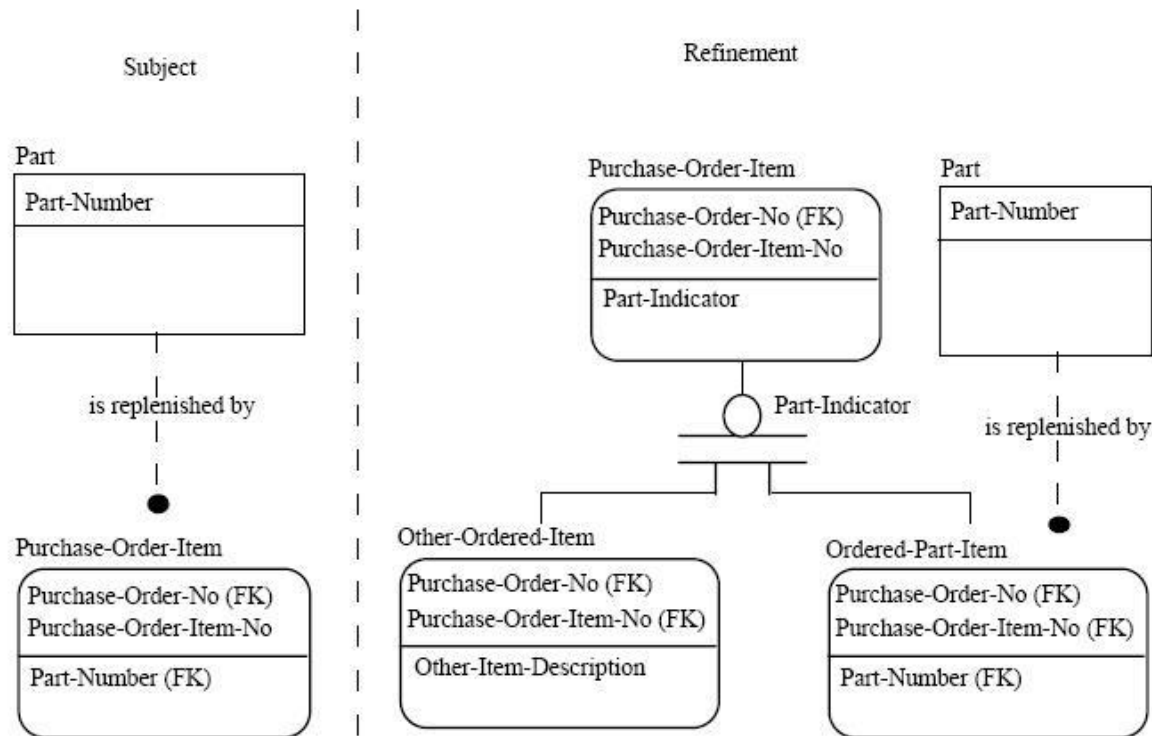
Entity-Relationship Matrix Example

	Buyer	Requester	Approver	Purchase Requisition	Purchase Req. Item
Buyer		X		X	
Requester	X			X	
Approver				X	
Purchase Requisition	X	X	X		X
Purchase Req. Item				X	

An Entity-Relationship Matrix only reflects that a relationship of some kind may exist.

IDEF1X

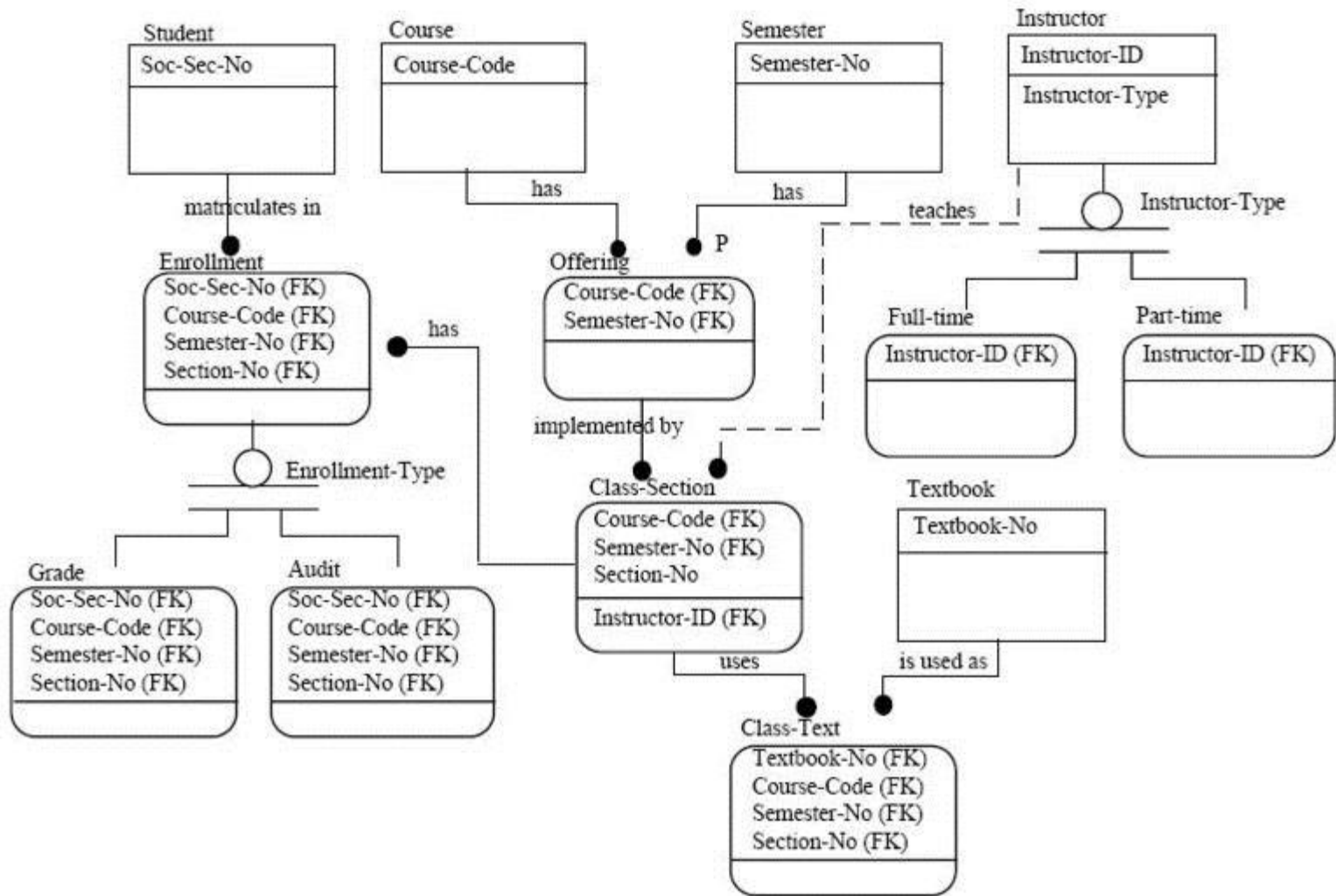
- ▶ Fáze III – definice klíčů
 - ▶ definice klíčů pro každou entitu
 - ▶ migrace primárních klíčů (PK) do cizích klíčů (FK)
 - ▶ validace relací a klíčů, upřesnění nspecifikovaných relací z Fáze II



This structure does not provide for purchase order items that may not be for parts (e.g., services, administrative supplies, etc.).

This structure does provide for purchase order items that may not be for parts (e.g., services, administrative supplies, etc.).

IDEF1X

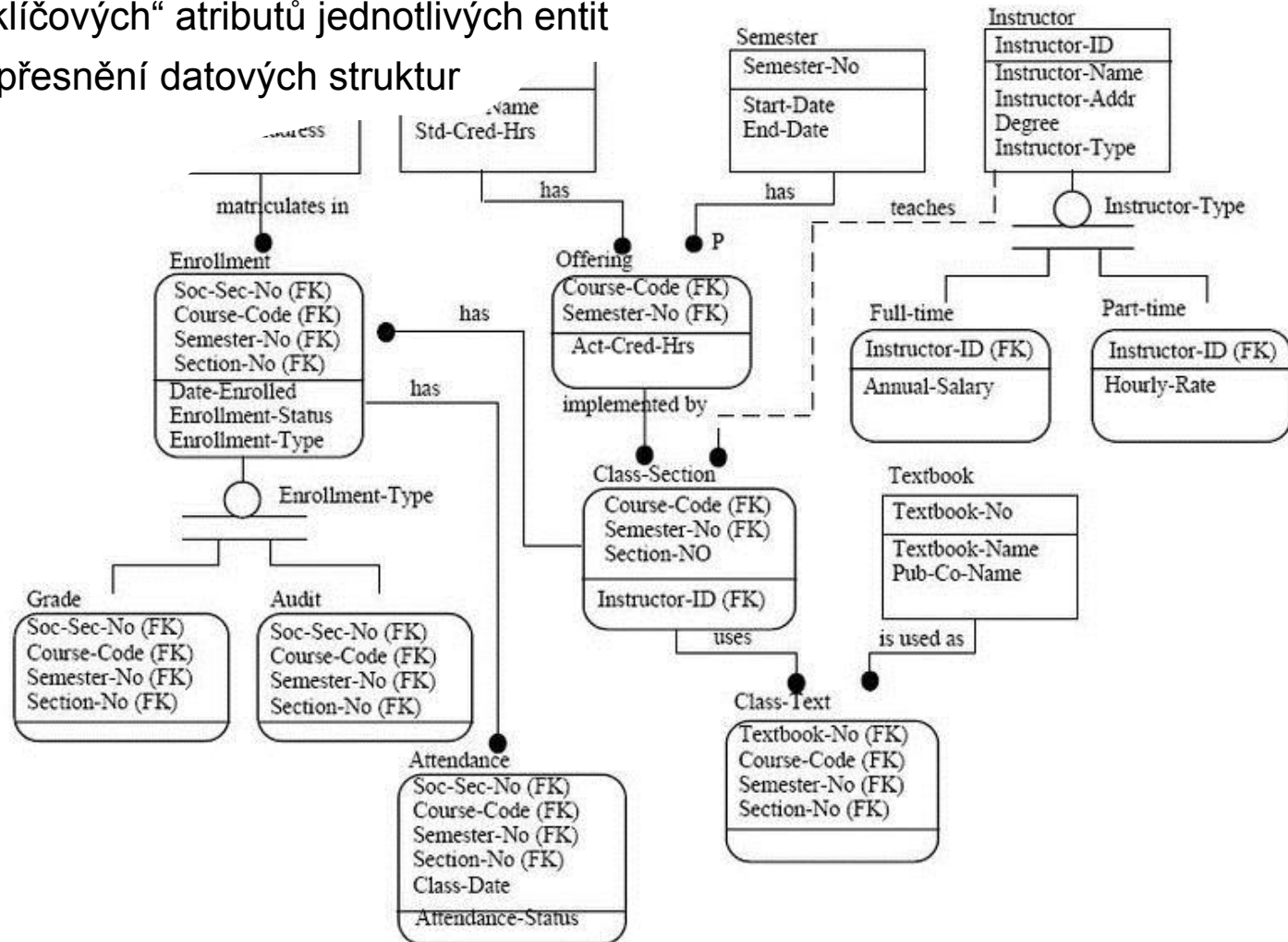


Příklad výstupu Fáze III – definovány PK a FK u všech relací



IDEF1X

- ▶ Fáze IV – **definice atributů** (závěrečná fáze)
 - ▶ identifikace všech atributů (vytvoření tzv. attribute pool)
 - ▶ přiřazení vlastnictví atributů
 - ▶ definice „neklíčových“ atributů jednotlivých entit
 - ▶ validace a upřesnění datových struktur



IDEF1 (information Model) vs. IDEF1x (Data Model)

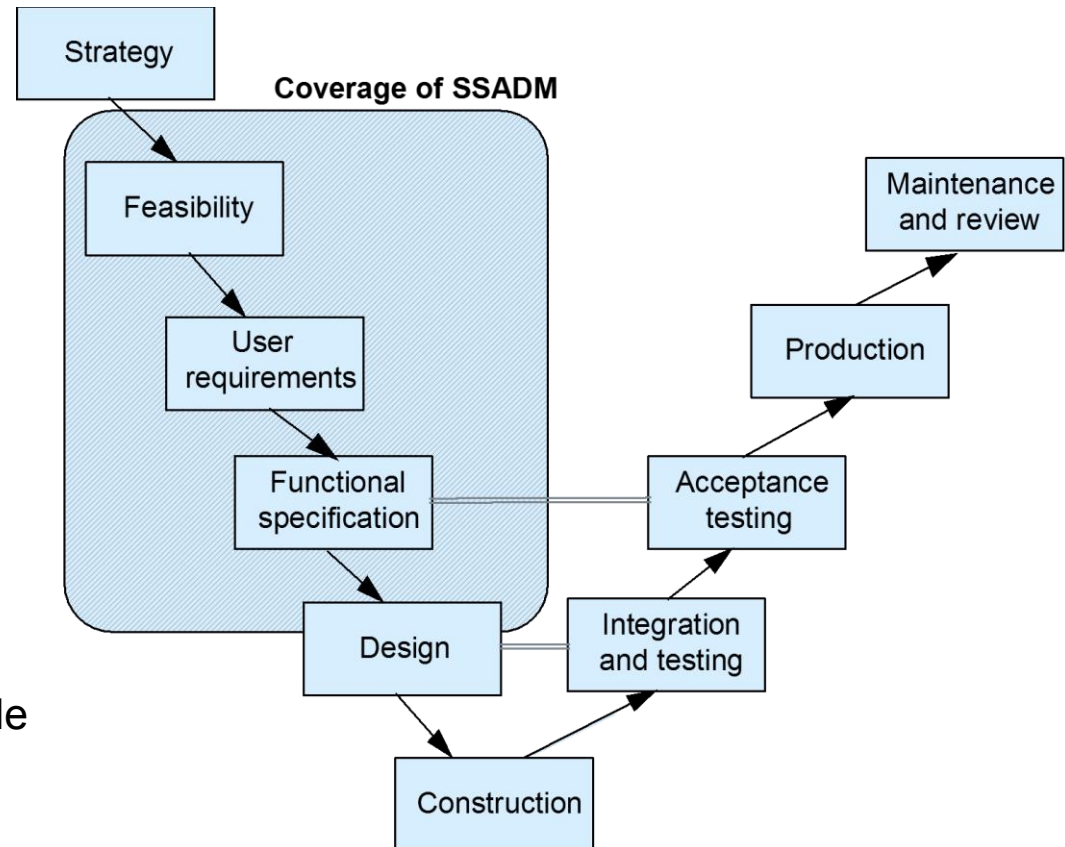
Information Model	Data Model
<p>focuses on:</p> <ul style="list-style-type: none">◆ Information collected, stored, and managed by the organization◆ Logical relationships within the organization reflected in the information	<p>focuses on:</p> <ul style="list-style-type: none">◆ Actual data elements in a relational database◆ Representation & structure of the data
<p>used for:</p> <ul style="list-style-type: none">◆ Problem identification◆ Requirements definition◆ Information system design	<p>used for:</p> <ul style="list-style-type: none">◆ Logical design of databases & applications◆ Physical design of database implementation

IDEF5 - Ontology Description Method

- ▶ An ontology is a domain vocabulary complete with a set of precise definitions or axioms that constrain the meanings of the terms sufficiently, to enable consistent interpretation of the data that use that vocabulary.
- ▶ General ontology construction steps:
 1. catalog the terms;
 2. capture the constraints that govern how those terms can be used to make descriptive statements about the domain; and
 3. build the model.

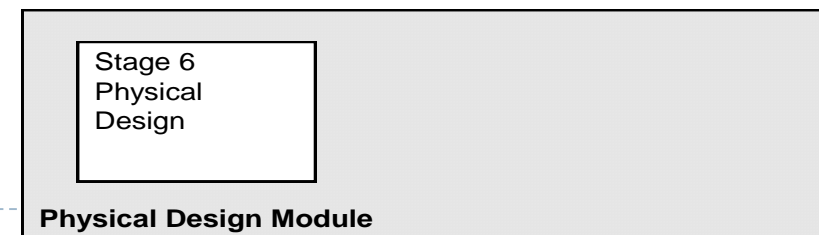
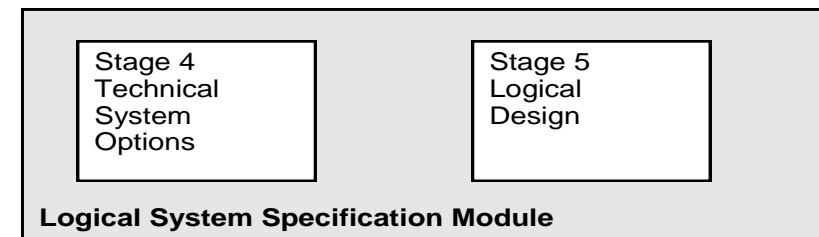
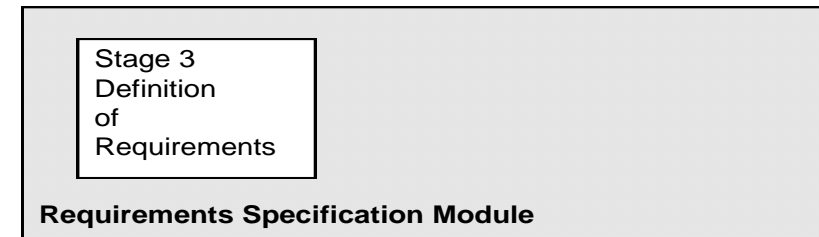
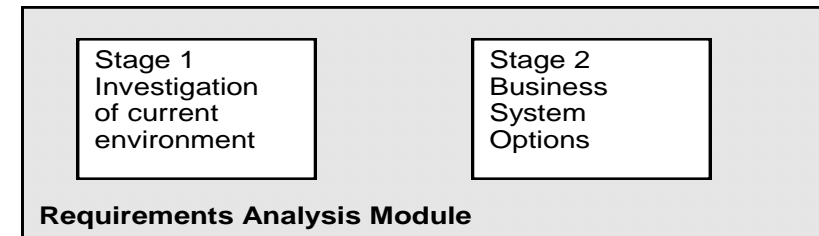
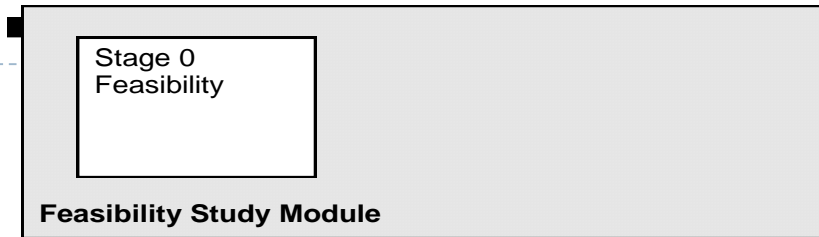
SSADM

- ▶ **SSADM (Structured System Analysis and Design Methodology)** bylo navrženo firmou Learmonth and Burchett Management Systems Plc. (UK) na základě vítězství v tendru vyhlášeném CCTA (Central Computing and Telecommunications Agency) a zavedeno r. 1981 jako **standardní metodologie pro vládní IT projekty v UK**; SSADM v.4 je otevřeným standardem, je volně dostupné i pro komerční využití.
- ▶ Hlavní aplikační doménou SSADM je **návrh IS**. Proto se zaměřuje zejména na datové toky, datové modely a životní cykly entit. **Důraz je kladen na detailní a strukturovaný přístup** v adekvátní etapě vývoje.
- ▶ Používá řadu nástrojů pro zachycení a vizualizaci detailů na každé úrovni návrhu.
- ▶ **Požadavky na nový systém jsou analyzovány spolu s poznatky o současném stavu** a jsou vytipována omezení, za kterých bude nový systém provozován.



SSADM – Hlavní složky

- ▶ **Strukturální model (Structural Model)** - soubor modulů, z nichž každý zahrnuje jednu až dvě etapy, přičemž každá etapa je určena posloupností kroků spolu s vstupy/výstupy a dalšími závislostmi
- ▶ **Techniky a postupy (Techniques and Procedures)**
 - ▶ **Modelování logických datových struktur (Logical Data Modeling)**
proces identifikace, modelování a dokumentace požadavků na datové struktury systému. Data jsou separována do entit, které ukazují, jaká informace je ukládána a jaké jsou vzájemné relace
 - ▶ **Modelování datových toků (Data Flow Modeling)**
proces identifikace, modelování a dokumentace toho, jak jsou data předávána v systému, jaké procesy je zpracovávají, jak jsou ukládána, co je zdroj a příjemce dat.
 - ▶ **Modelování chování entit (Entity Behavior Modeling)**
proces identifikace, modelování a dokumentace toho, jaké události ovlivňují jednotlivé entity, v jakých posloupnostech tyto události nastávají a jak ovlivňují životní cykly entit.
- ▶ **Slovník SSADM (Dictionary)** - definuje veškeré výstupy metody a příslušná kvalitativní kritéria
- ▶ **Projektové postupy (Project Procedures)** - další prováděné relevantní aktivity, které však nejsou přímo součástí SSADM



SSADM – Mapování na vývojový proces

Fáze projektu	Hlavní kroky procesu	Etapa SSADM
Project Request	Initial Request Statement (IRS)	-
Feasibility Study	Feasibility Study (FS)	Stage 0 (Feasibility)
System Analysis and Design	System Analysis (SA)	Stage 1 (Investigation of Current Environment) Stage 2 (Business System Options) Stage 3 (Definition of Requirements) Stage 4 (Technical System Options)
	Logical System Design (LSD)	Stage 5 (Logical Design)
Implementation	Physical System Design (PSD)	Stage 6 (Physical Design)
	Programming Development (PD)	-
	System Integration & Testing (SI&T)	-
	User Acceptance Testing & Training (UA&T)	-
	System Installation & Production (SI&P)	-
	Project Evaluation Review	-
Post Implementation Review nad Maintenance	Post Implementation Review	-
	Maintenance	-



FEASIBILITY STUDY MODULE

▶ Studie proveditelnosti, SSADM Stage 0.

▶ Účel:

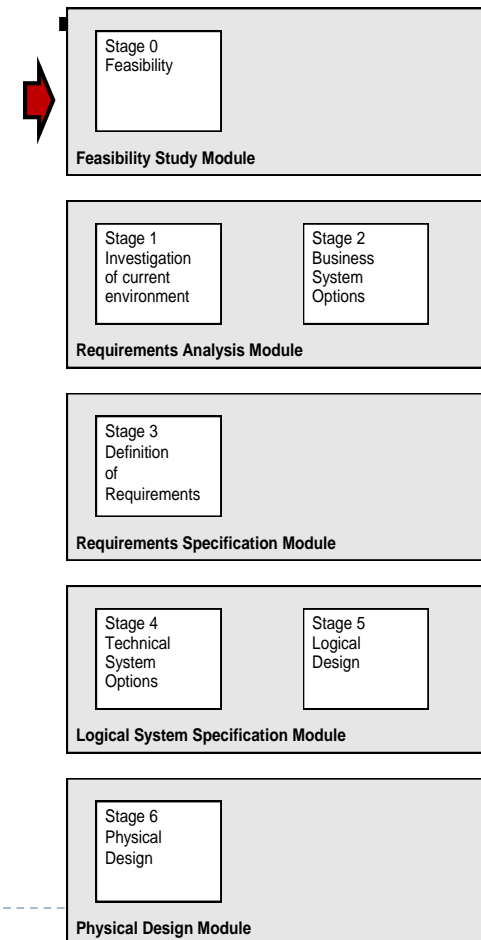
- ▶ rozhodnout o tom, zda je projekt reálný;
- ▶ definovat zhruba rámec a cíle projektu.

▶ Otázky, na které odpovídá:

- ▶ Je reálné, že systém splní očekávané cíle?
- ▶ Vybáží přínosy systému vynaložené náklady?
- ▶ Jak rozsáhlý systém je potřeba?

▶ Výstupy:

- ▶ studie proveditelnosti



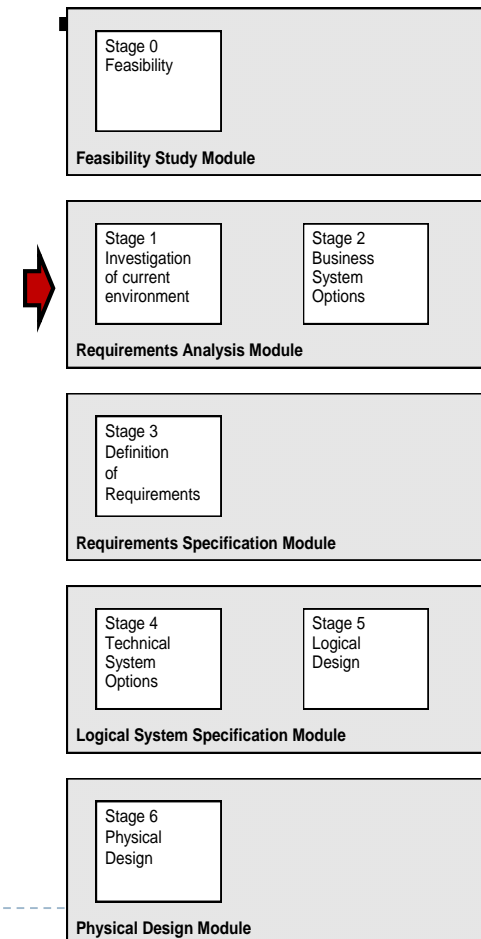
SSADM - Popis modulů

REQUIREMENTS ANALYSIS MODULE

- ▶ **Analýza požadavků, SSADM Stage 1 a 2**
- ▶ *Účel:*
 - ▶ přehledově definovat požadavky na nový systém na základě stávajícího systému (Stage 1)
 - ▶ dokumentovat další požadavky uživatele na nový systém (Stage 2)
- ▶ *Přehled:*

Požadavky jsou analyzovány v následujících posloupnostech:

 - ▶ analýza současné funkce, dat a kategorií uživatelů systému;
 - ▶ zjištění požadavků na nový systém;
 - ▶ formulace základních rysů systému.
- ▶ *Výstupy:*
 - ▶ analýza požadavků



SSADM - Popis modulů

REQUIREMENTS SPECIFICATION MODULE

► Specifikace požadavků, SSADM Stage 3.

► Účel:

- detailně specifikovat požadavky na nový systém

► Přehled:

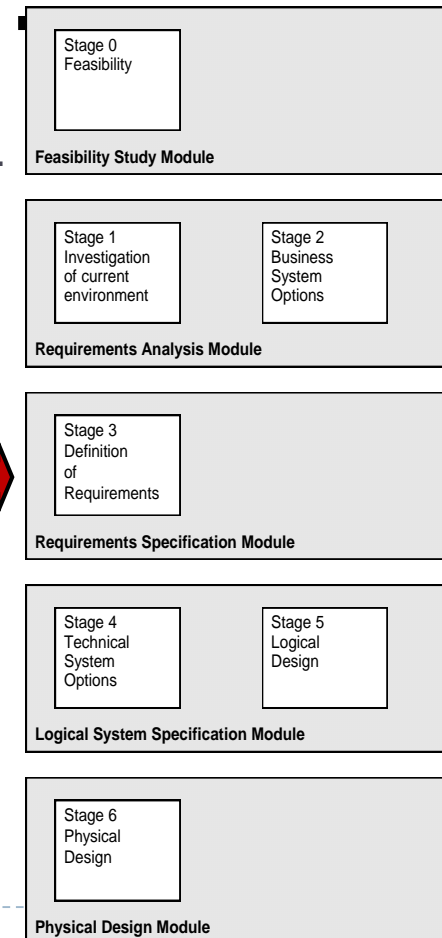
- V tomto modulu se konkretizují a kvantifikují požadavky na systém, vzešlé z předchozí analýzy. Data a funkce jsou popsány použitím několika technik z různých hledisek. Využívá prototypování (zejména uživatelského rozhraní).

► Výstupy konečné:

- katalog dat (Data Catalogue);
- katalog požadavků (Requirements Catalogue);
- specifikace zpracování (Processing Specification).

► Mezivýstupy

- DFD požadovaného systému (Required System Data Flow Model);
- E/E matice (Event/Entity Matrix);
- relační model (Relational Data Analysis Working Paper);
- uživatelské role v systému (User Roles);
- prototypy (Prototyping Products).



SSADM - Popis modulů

LOGICAL SYSTEM SPECIFICATION MODULE

▶ Logický návrh systému, SSADM Stage 4 a 5

▶ Cíl:

- ▶ popsat z technického hlediska prostředí nového systému (Stage 4);
- ▶ poskytnout detailní specifikaci průběhu zpracování v systému a požadavky na interakci s uživateli (Stage 5).

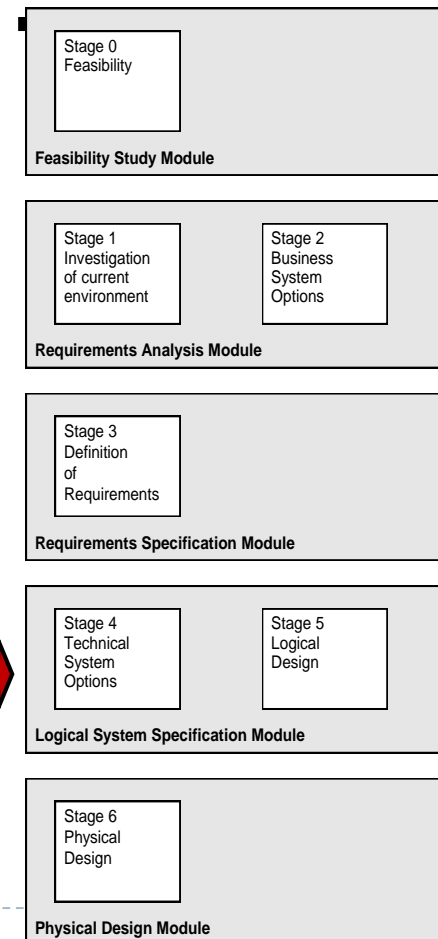
▶ Přehled:

Dvě paralelní větve:

- ▶ výběr, zhodnocení a zdokumentování technického okolí systému;
- ▶ detailní a strukturovaný popis procedur a uživatelských dialogů.

▶ Výstupy:

- ▶ zvolené technického řešení (Selected Technical System Option);
- ▶ popis technického okolí systému (Technical Environment Description);
- ▶ logický návrh (Logical Design).



SSADM - Popis modulů

PHYSICAL DESIGN MODULE

► Fyzický návrh systému, SSADM Stage 6

► Účel:

- slouží jako spojovací článek mezi logickým návrhem a vlastní implementací systému.

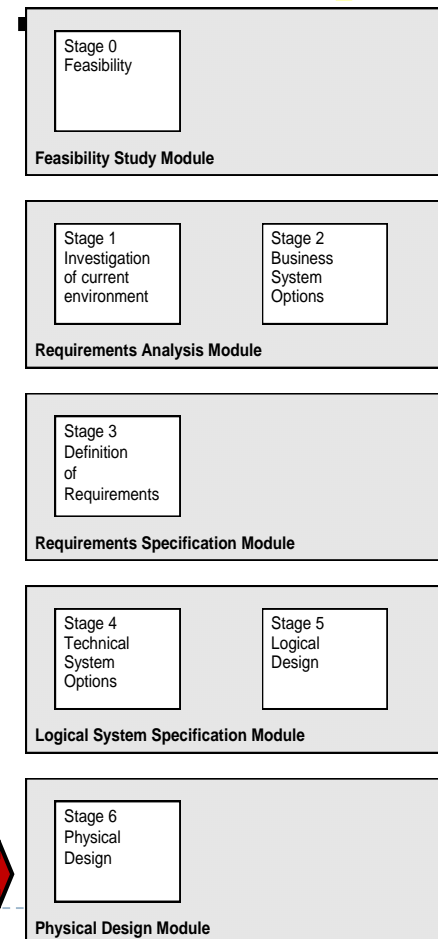
► Přehled:

Vstupy pro tento modul jsou z různých zdrojů:

- Funkcionalita systému, datové popisy atd. jsou odvozeny z SSADM dokumentace.
- Konkrétní metody implementace závisí na HW a SW
- Obecné zásady pro vývoj a systémovou dokumentaci.

► Výstupy:

- fyzický návrh systému (Physical Design);
- standardy pro vývoj aplikace (Application Development Standards);
- fyzický popis prostředí (Physical Environment Specification).



Modelování datových toků

Data Flow Model (DFM) se používá v několika etapách SSADM, a to:

- ▶ současný fyzický DFM (Current Physical Data Flow Model);
- ▶ logický DFM (Logical Data Flow Model);
- ▶ DFM požadovaného systému (Required System Data Flow Model).

DFM se skládá z:

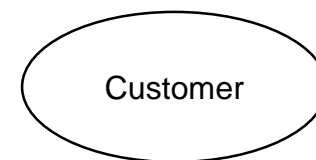
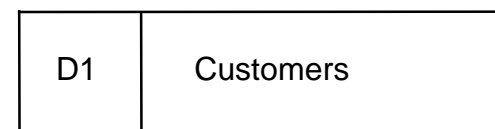
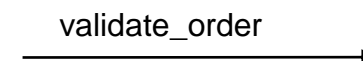
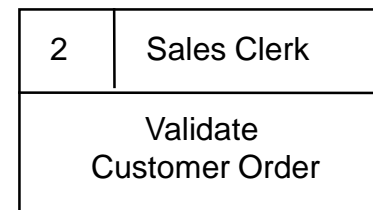
- ▶ DFD vrstvené do úrovní (Data Flow Diagrams);
- ▶ popis elementárních procesů (Elementary Process Descriptions);
- ▶ popis vnějších entit (External Entity Descriptions);
- ▶ popis vstupů/výstupů (I/O Descriptions).

Diagramy datových toků (DFD)

- ▶ V grafické formě popisují toky dat v systému a interakci s okolím. Specifikují zdroje a cíle dat, jejich transformace (hrubě) a identifikují místa, kde se data "skladují" (databáze). Slouží též k identifikaci hranice systému (zdroje či cíle dat mohou být vně systému).

DFD se skládá z následujících elementů:

- ▶ **Procesy**, které reprezentují transformaci nebo manipulaci s daty. Identifikátor procesu je číslo.
- ▶ **Datové toky**, což jsou kanály mezi ostatními elementy, kterými mohou proudit předdefinované množiny dat. Datové toky na spodní úrovni diagramu jsou popsány svým obsahem.
- ▶ **Datové sklady**, obsahující uložená data. Mohou být tzv. hlavní (main), které jsou persistentní, nebo přechodné (transient), ukládané dočasně. Identifikátor datového skladu má syntaxi 'D' plus číslo.
- ▶ **Externí entity**, což jsou zdroje nebo příjemci dat vně systému. Identifikátorem externí entity je malé písmeno.

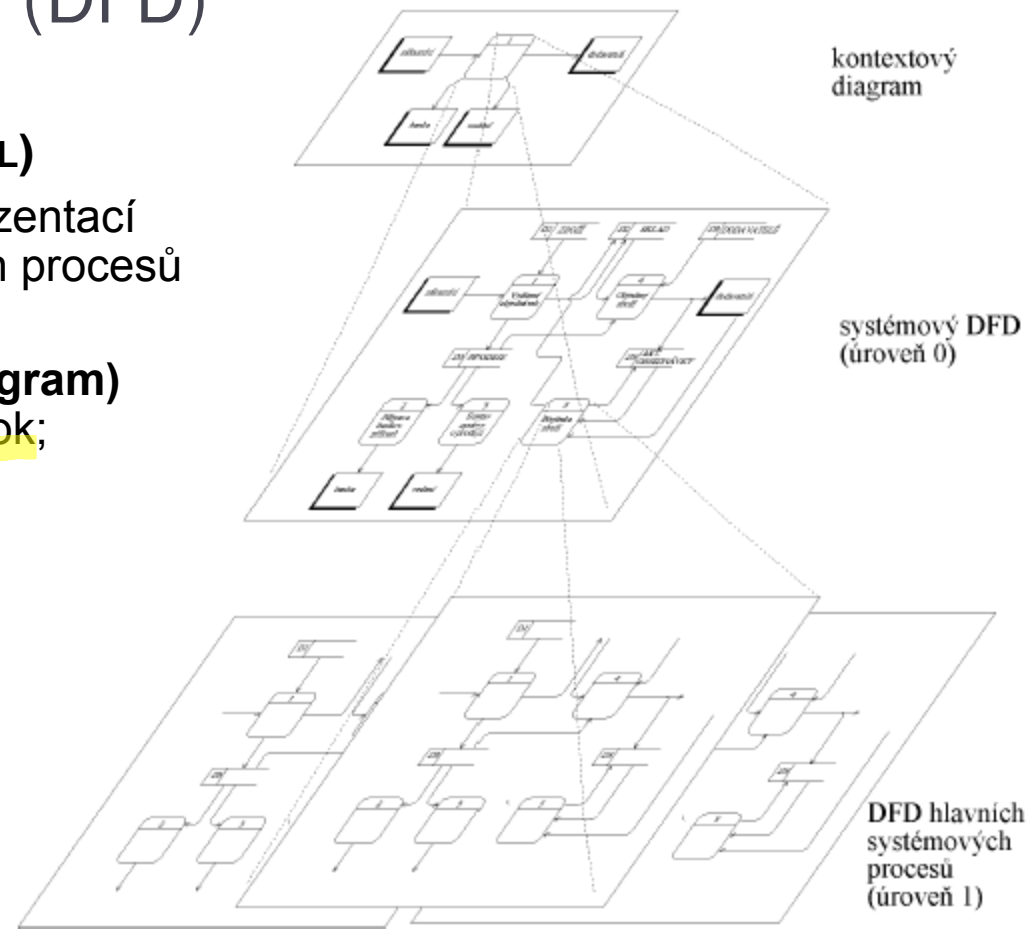


Diagramy datových toků (DFD)

SOUČASNÝ FYZICKÝ DFM (CURRENT PHYSICAL DATA FLOW MODEL)

Současný fyzický DFM model je reprezentací datových toků a s nimi souvisejících procesů v současném systému:

- ▶ **kontextový diagram (Context Diagram)**
- uvažuje celý systém jako jeden blok;
- ▶ **diagram toku dokumentů (Document Flow Diagrams)**
– popisuje, jak v současném systému putují dokumenty a formuláře a jaký je jejich smysl;
- ▶ **diagram toků zdrojů (Resource Flow Diagrams)**
– specifikuje toky "hmoty" spíše než informace.

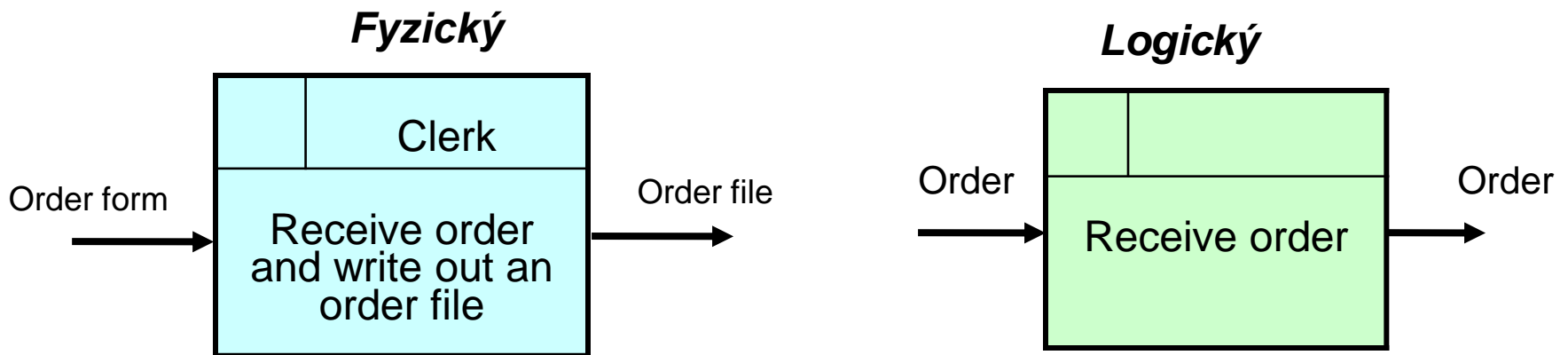


DFM POŽADOVANÉHO SYSTÉMU (REQUIRED SYSTEM DATA FLOW MODEL)

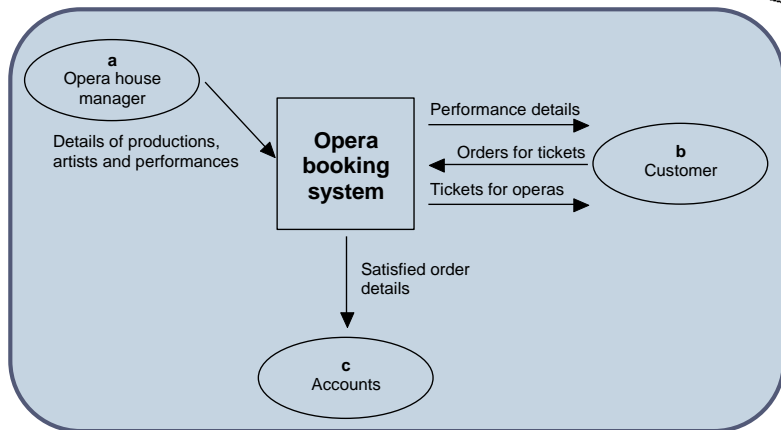
- ▶ DFM požadovaného systému je založen na specifikaci funkce nového systému. Obvykle se skládá z kontextového diagramu, diagram toku dokumentů (Document Flow Diagrams) a diagram toků zdrojů (Resource Flow Diagrams).

Diagramy datových toků (DFD)

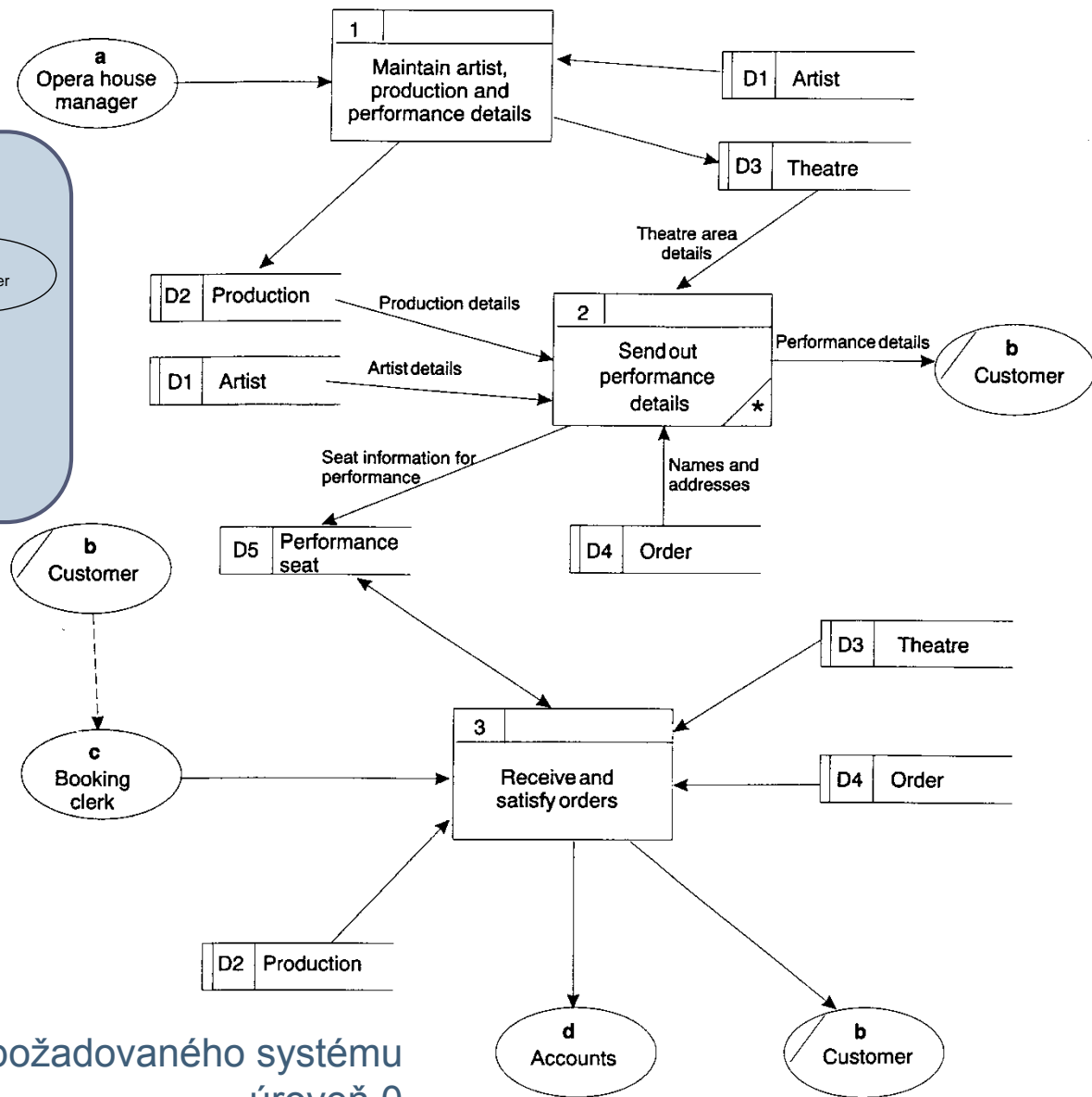
- **Fyzický DFD** reprezentuje, **JAK se co děje**.
Obsahuje popis problému, **nezávisí na aktérech a je implementačně nezávislý**.
Má tendenci obsahovat redundantní datové sklady / zpracování.
Má tendenci používat jména oddělení podniku, lidí, formulářů, zařízení.
- **Logický DFD** je **extrakcí fyzického DFD**. Je to **logická reprezentace** systému, která představuje, **CO systém dělá**.
Je **implementačně nezávislá**. Zaměřuje se **pouze na datové toky** mezi procesy, bez ohledu na specifická zařízení, úložiště aj.



DFD - příklad

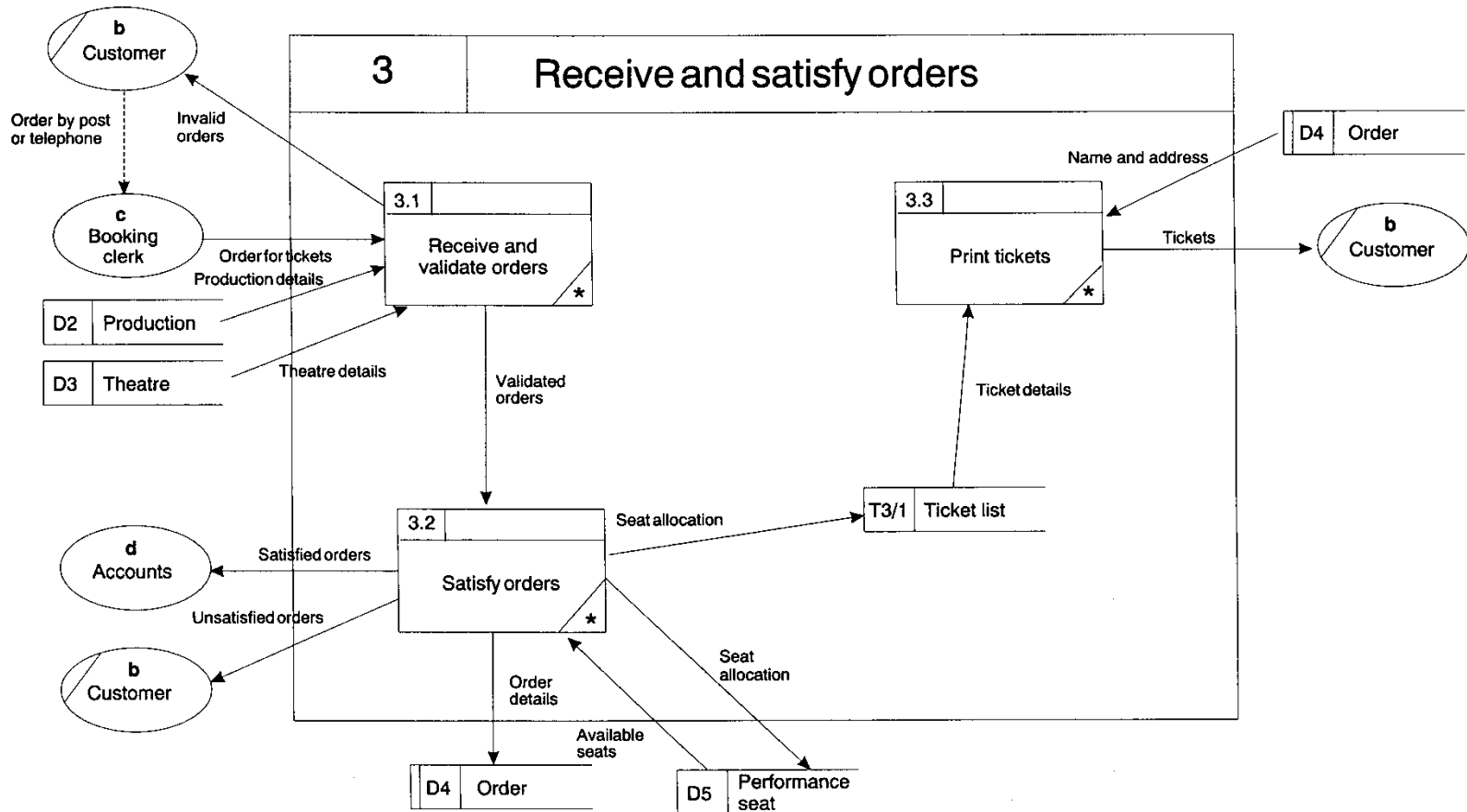


Kontextový diagram

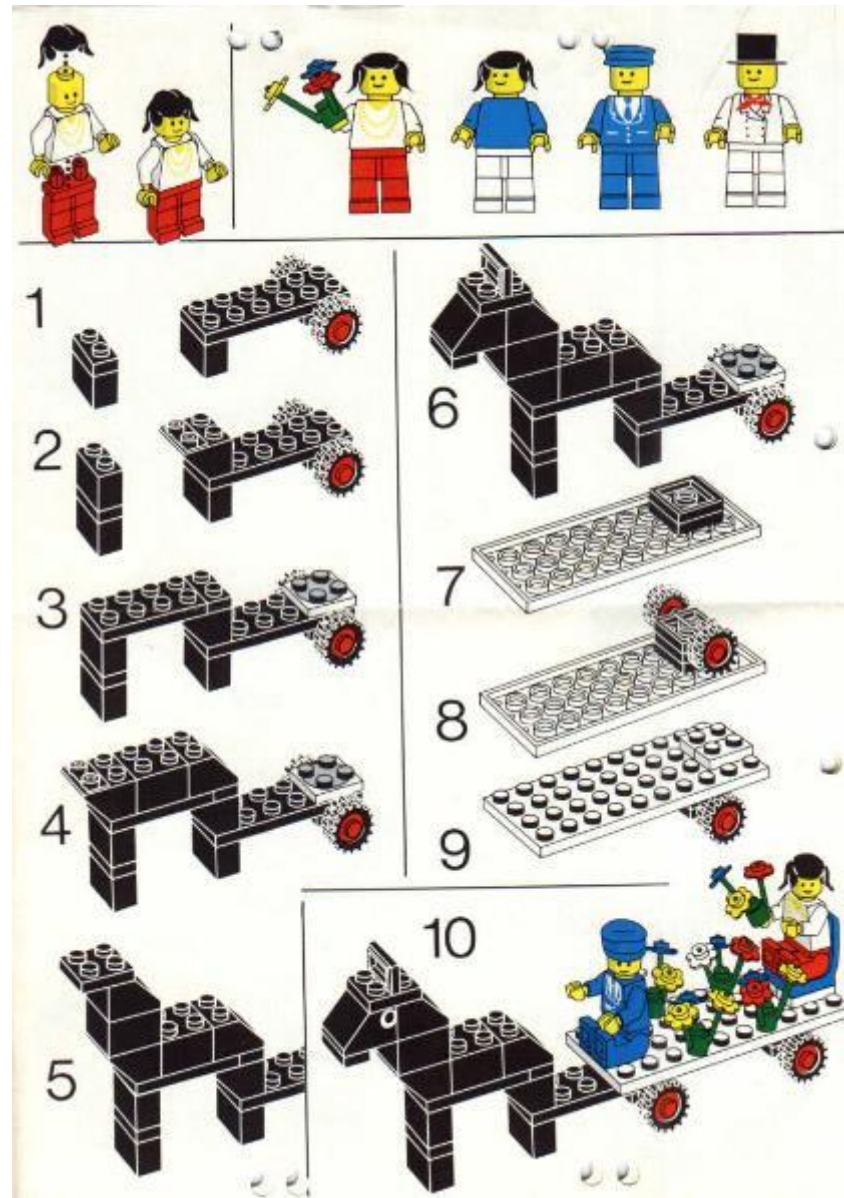


DFD požadovaného systému
úroveň 0

DFD - příklad



„Optimální“ Data Flow Diagram?

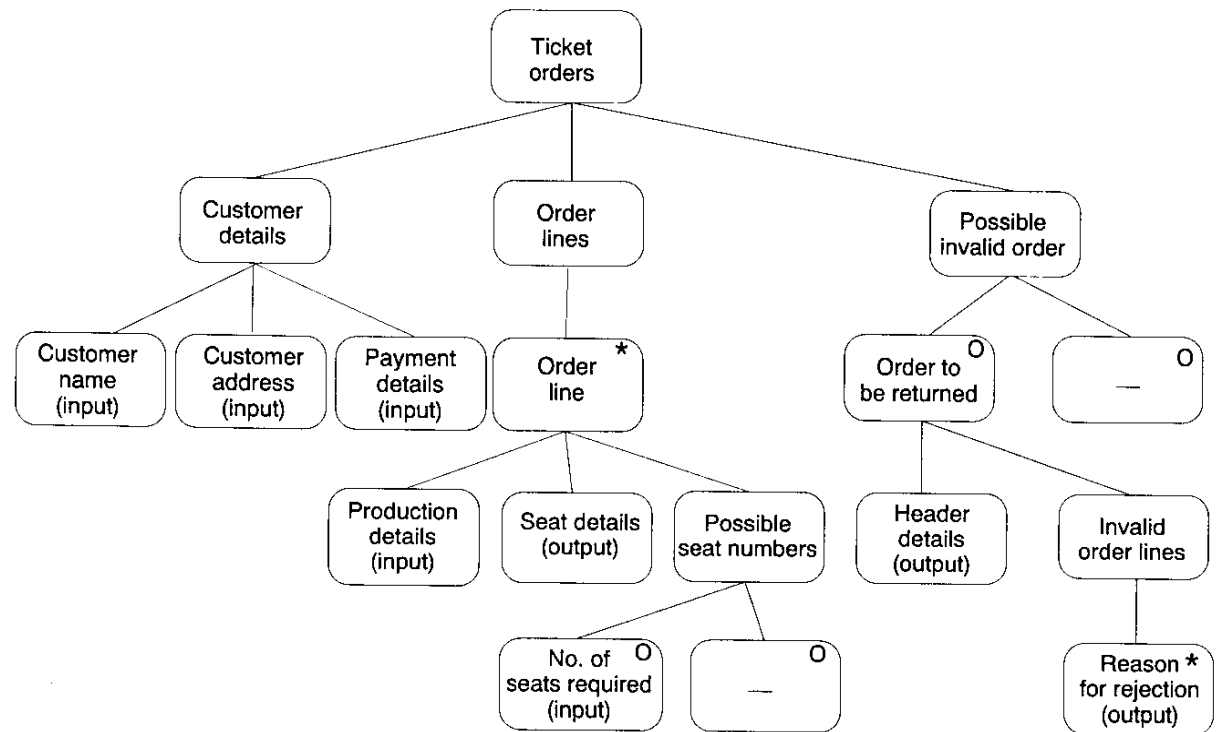


I/O Description - příklad

- ▶ **I/O struktury** popisují vstupy/výstupy funkcí a jsou součástí **podpůrné dokumentace** funkcí. Datové položky jsou strukturované. Tyto diagramy se využívají jako **základ pro návrh dialogů** v modulu *Logical System Specification Module*.

Pro **každou funkci se identifikuje jedna I/O struktura**, a to obvykle z popisu vstupů/výstupů funkce (I/O Descriptions).

- ▶ **Příklad I/O diagramu** relevantnímu funkci 'receive orders for tickets' z úlohy rezervace vstupenek



I/O Description - příklad

From	To	Data flow name	Data content	Comment
c	3.1	Order for tickets	Customer name Customer address Date of performance Theatre area No. of seats required Price per seat Total price for each seat Total for the Whole order Method of payment Credit card no. Expiration date	A number of tickets can be on one order
3.1	b	Invalid orders	Customer name Customer address Date of performance Theatre area No. of seats required Price per seat Total price for each seat Total for the Whole order Method of payment Credit card no. Expiration date	The original order is sent back with a note detailing the errors
3.2	d	Satisfied orders	Customer name Customer address Date of performance Theatre area No. of seats satisfied	For each of the order lines satisfied
3.2	b	Unsatisfied orders	Customer name Customer address	

I/O Structure name: Ticket orders			4-1
Data flows represented:			c-3.1 3.1-b
I/O Structure Element	Data item	Comments	
Customer name	Customer name		
Customer address	Customer address		
Production details	Date of performance Theatre area		
Seat details	No. of seats Price per seat Total		
No. of seats required	No. of seats		
Header details	Customer name Customer address		
Reason for rejection	Reason for rejection	Internally generated msg	

Každá I/O struktura je dokumentována popisem (I/O Structure Description) ve formě tabulky

Matrice událost/entita (Event/Entity Matrix)

E/E matice je výchozím bodem pro *Entity Lifecycle History (ELH)* analýzu.

Poskytuje křížový pohled na entity z logického datového modelu požadovaného systému a události, zdokumentované ve Funkční specifikaci. Pro uvedený příklad:

Event / Entity	Artist	Artist in performance	Bookable seat	Opera performance	Order	Order line	Part of theatre	Production	Seat at performance
Arrival of order					C	C			M
Details sent to credit company					M				
Credit card approved					M				
Request for new details sent					M				
New details received					M	M			
Tickets sent					M	D/M			
Performance takes place					M				
Change to personal details					M				
Retention period expires					D	D			
Receive new production details	C	C			M			C	
Performance details finalised	M/D	M	C	C					C
Opera house details changed	M	M	M	M			C/M	M	M

Pro každý element v E/E matici: entita je událostí *M=Modified, C=Created, D=Deleted*

Matrice se užívá ke kontrole, že:

- všechny entity vznikají alespoň jednou událostí,
- každá událost ovlivňuje alespoň jednu entitu.



Popis života entit (ELH)

- ▶ **Popis života entit (*Entity Life Histories*)**
- ▶ **Účel**
- ▶ Určuje (zpravidla ve formě incidenční tabulky), které události (přicházející zvenčí systému nebo významné časové okamžiky) ovlivňují které entity či jejich datové reprezentace. Zpravidla se uvádějí i požadavky na priority zpracování při souběhu více událostí.
- ▶ Analýza životního cyklu entit (ELH) se uskutečňuje technikou zvanou *entity/event modelling*. Smyslem je nalézt příčiny změn v datech v systému, definovat jejich omezení a modelovat přesný vliv na jednotlivé datové elementy.
- ▶ ELH analýza je technika založená na stromových diagramech a zachycuje životní cyklus entity od jejího vzniku až po zánik, reflektuje veškeré události, které mohou entitu pozměnit, a zachycuje jejich časový sled.
- ▶ Výchozím materiálem pro tvorbu je E/E matice. Pro každou entitu se vytváří vlastní diagram, obsahující stromovou strukturu, kde kořenem je zvolená entita, listy reprezentují efekt identifikovaných událostí a mezilehlé uzly definují strukturu.

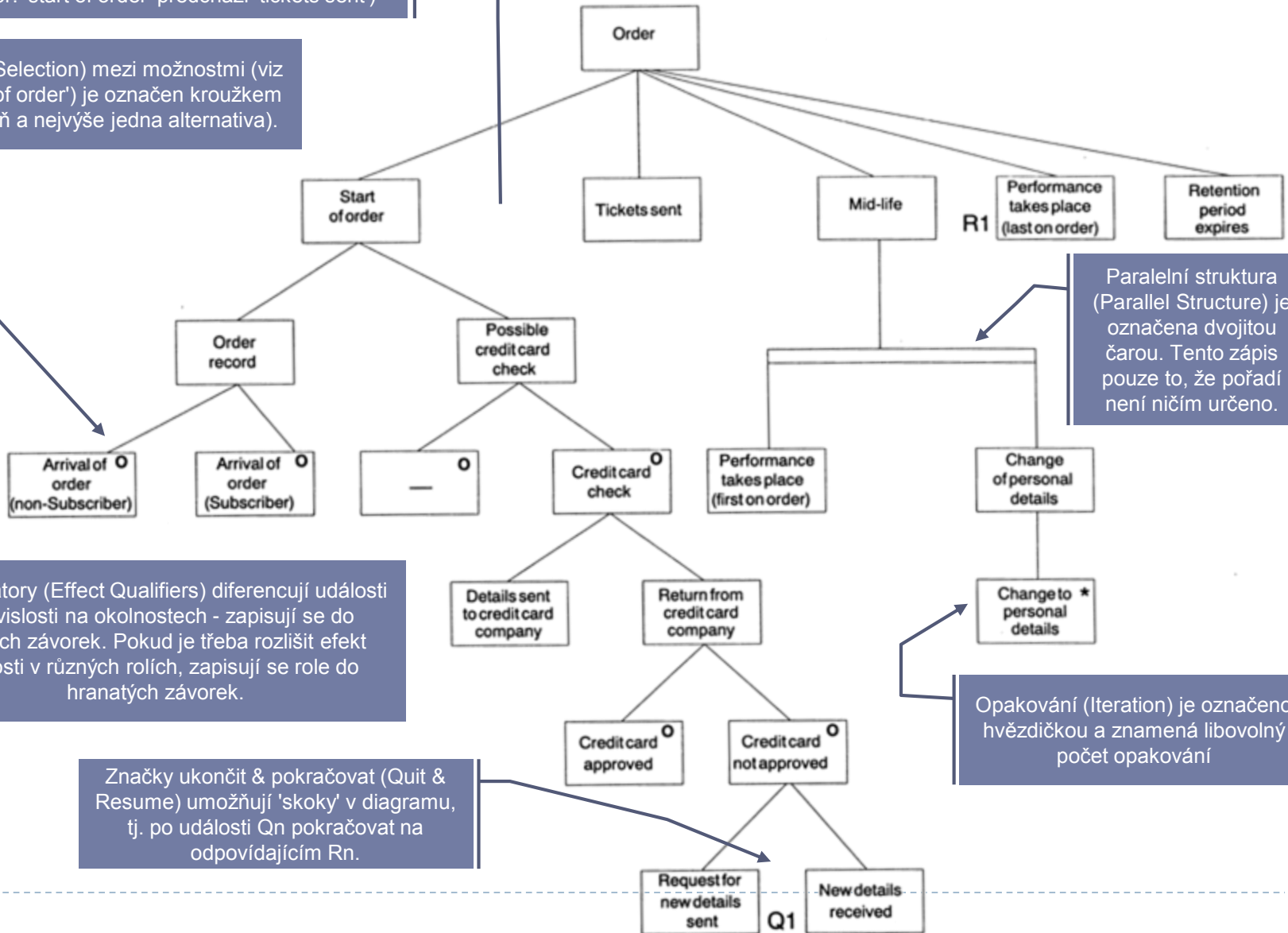
Popis života entit (ELH)

Posloupnost (sequence) je rozlišena horizontální úrovní (např. 'start of order' předchází 'tickets sent')

Výběr (Selection) mezi možnostmi (viz 'arrival of order') je označen kroužkem (alespoň a nejvýše jedna alternativa).

Kvalifikátory (Effect Qualifiers) diferencují události v závislosti na okolnostech - zapisují se do kulatých závorek. Pokud je třeba rozlišit efekt události v různých rolích, zapisují se role do hranatých závorek.

Značky ukončit & pokračovat (Quit & Resume) umožňují 'skoky' v diagramu, tj. po události Qn pokračovat na odpovídajícím Rn.



Paralelní struktura (Parallel Structure) je označena dvojitou čarou. Tento zápis pouze to, že pořadí není ničím určeno.

Opakování (Iteration) je označeno hvězdičkou a znamená libovolný počet opakování

Výběr správné metodiky?

- ▶ Existují i další metodiky jako UML.
- ▶ Nelze jednoznačně rozhodnout, která metodika je nejlepší.
- ▶ Srovnání různých metodik ukazuje, že jimi lze popsat totéž, ale jiným způsobem.
- ▶ Často kombinace různých přístupů.
- ▶ Rozhoduje poptávka a zkušenost dodavatele.

MALÁ ODBOČKA – ENTERPRISE ENGINEERING

MODELOVÁNÍ DEMO

Enterprises are social systems - General Systems Theory



- **Construction perspective**

- Operation
- Governance, Risk (GRC)
- Efficiency & Effectiveness
- Operational control



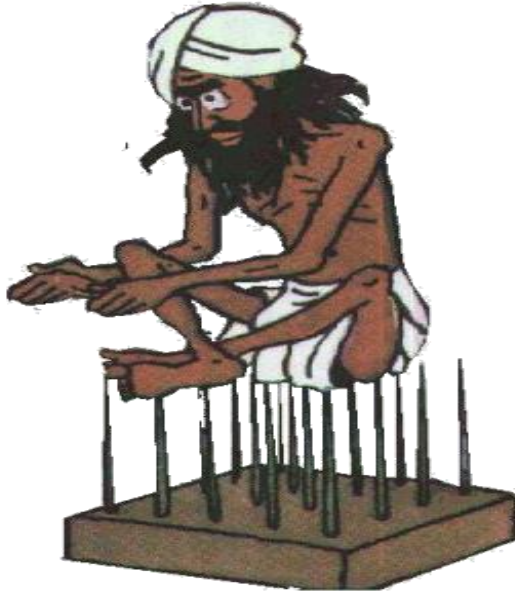
- **Function perspective**

- Markets & customers
- Compliance to regulations (GRC)
- Competition
- Agility; fast & unpredictable adaptations

Enterprise IT systems are systems that support enterprises and their operation:

- *Descriptive* IT systems – provide some functional perspective on enterprises (accounting, HRM, resource, etc).
- *Prescriptive* IT systems – enforce operational compliance of humans in an enterprise to a model (~workflow, BPM)
- *Production* IT systems – support the production (~ production tools, ~ERP)

Problem domain: IT systems for enterprises



Life is full of suffering:

- 1 Business-IT alignment fails
- 2 IT system engineering is a #@\$\$
- 3 IT systems resist evolution

IT state of the art fails too often: Zachman, BPMN, BPEL, EPC, TOGAF, Flow-charts, ARIS...(extensive literature)

Or do software engineers fail?

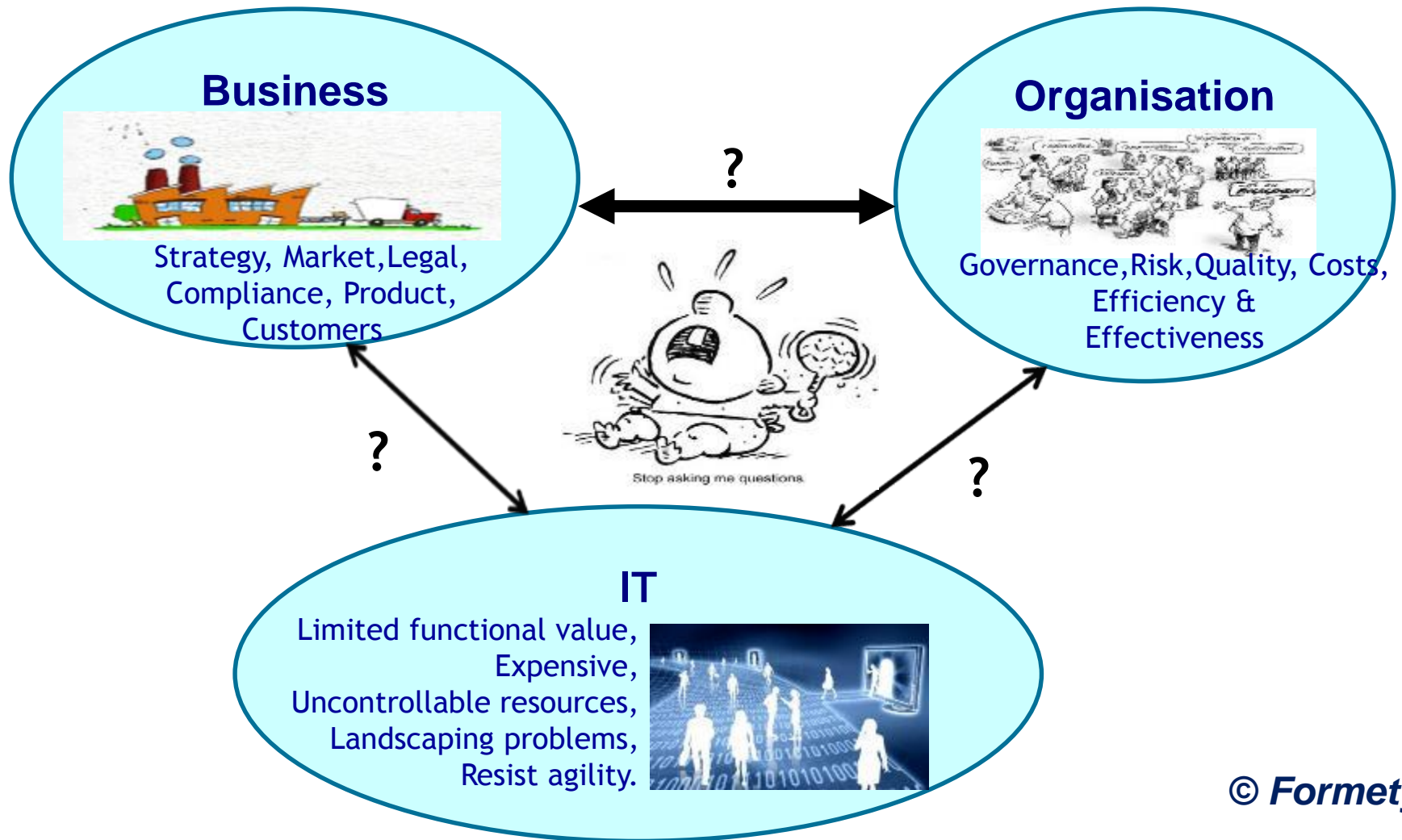
Symptoms or causes or deeper causes?

Questions regarding construction of enterprise IT systems:

How to construct *better* information systems? - function perspective

How to construct information systems in a *better way*? - construction perspective

Problem Domain: 2- Mismatch of Domains and their Concepts



© *Formetys*

Unmanageable complexity; unclear domains, concepts and relations; bad methods

Problem: State of the art modeling methods fail

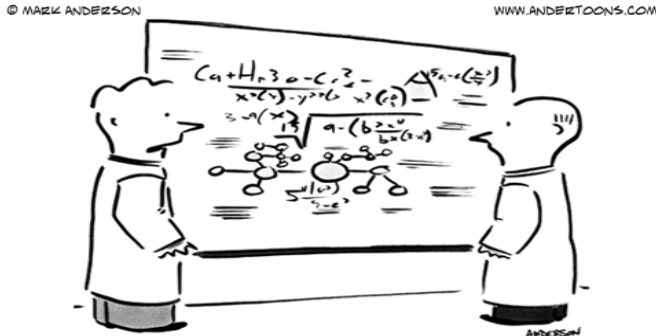


Best practice methods:

“we don't understand why this seems to work well, but we try to repeat this, while we don't know how to do it”.



Unsubstantiated promises followed by catastrophes; Nobody questions the usual “consultants, management & sales BS”



“I don't know, it's a little formulaic.”

Lack of scientific foundations,
Lack of validation, verification,
Lack of formal rigor.

Better than nothing - not good enough

Observations and Conclusions

State of the art enterprise IT systems fail too often

but

Programmers are NOT guilty,
'Business consultants' fail to provide high quality specifications;

because

The application domain is not well 'understood';
Unmanageable complexity,
unclear domains, unclear concepts, unclear relations between concepts;

therefore

Best practices, weak tools and management BS are state of the art;

It all boils down to

Lack of appropriate scientific theories, formal methods, engineering science.

An Approach to Mitigate the observed Problems

The Enterprise Operating System (EOS)

well founded on the

Discipline of Enterprise Engineering

based on

Understanding the domain - empirical sciences

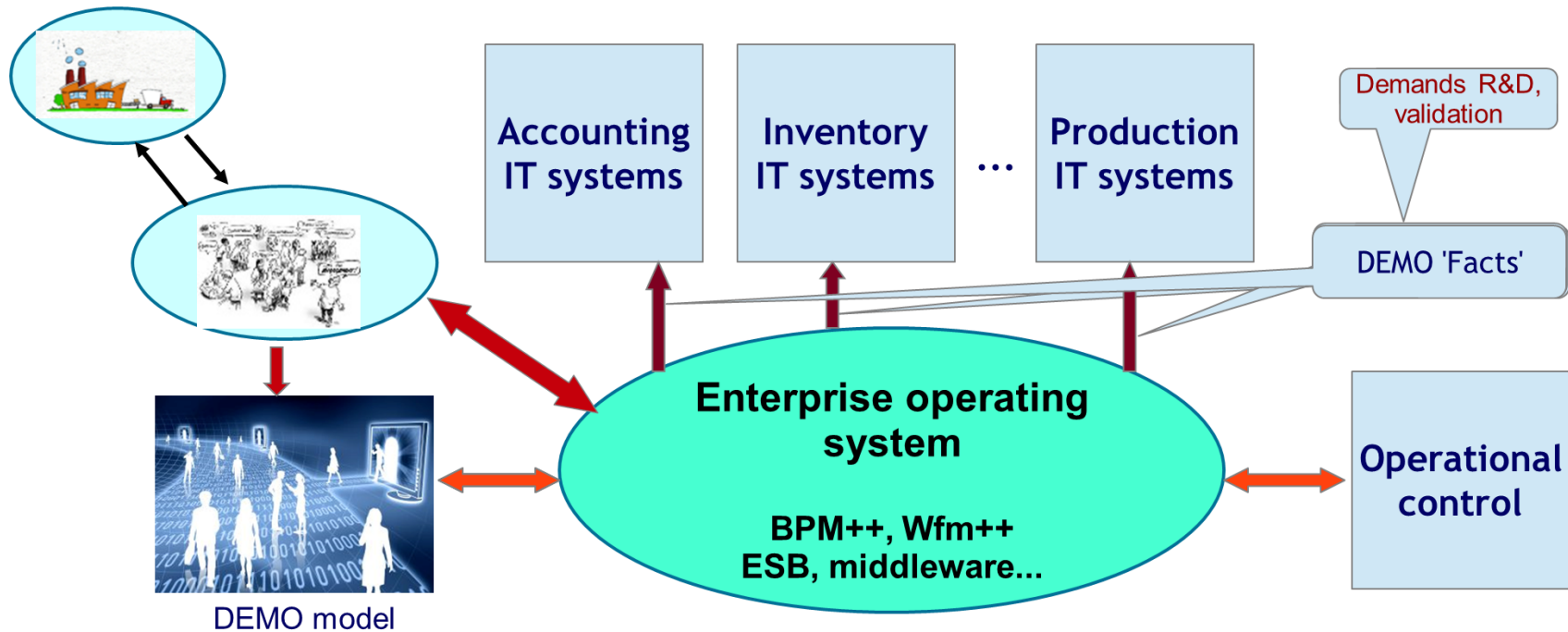
High quality specifications - formal methods

Appropriate methods - engineering sciences

and a

Human Centered Approach

EOS - Support for Enterprise IT Systems



Descriptive IT systems: IT systems; provide a functional perspective on operation (accounting, personnel, inventory ...)

Production IT systems: Support the production (parts) of an enterprise (document production, case management, ERP ...)

Facts: Represent phenomena in reality - production results (specified by the DEMO State Model)

State of the art silo type IT systems now share a common foundation, (may) eliminate (largely) ESB, middle ware, via abstractions and complexity reduction(*).

* Demands substantial further R&D and empirical validation

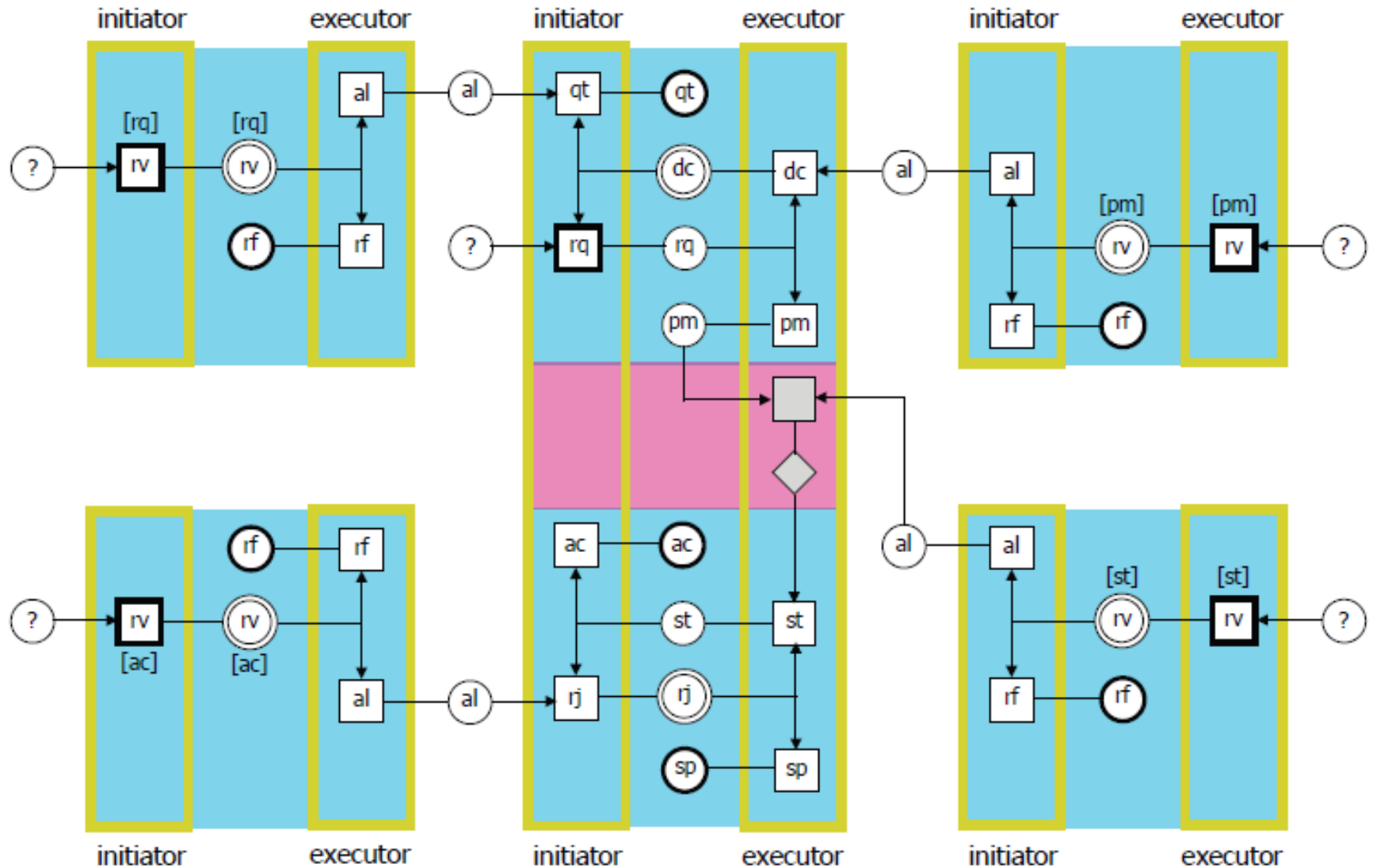
© **Formetys**

WHY DEMO?

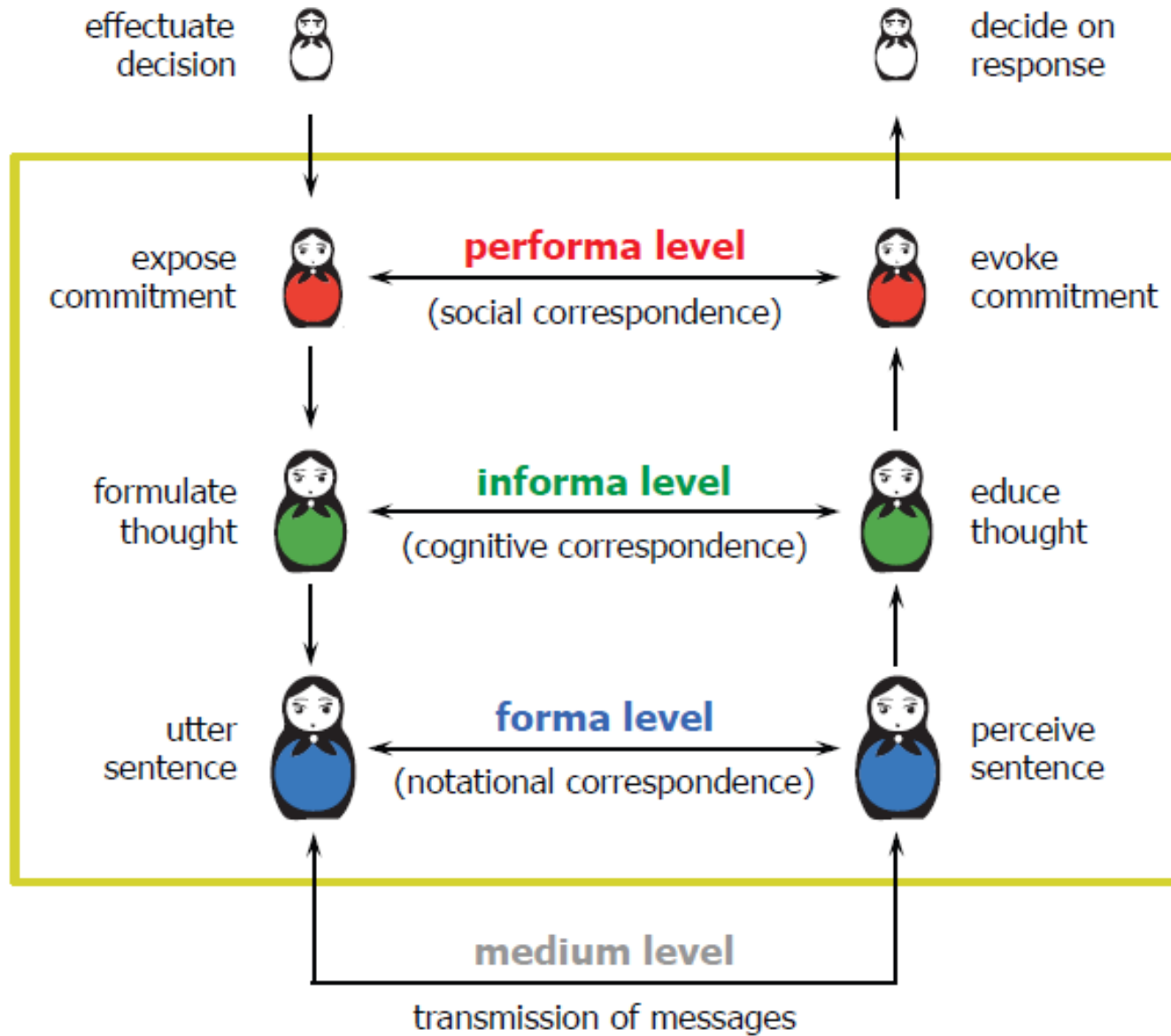
- ▶ It Looks for an Essence of an Organization
- ▶ Transaction Pattern
- ▶ Coordination Act
- ▶ Three Aspects of Organization



TRANSACTION PATTERN



COORDINATION ACT



THREE ASPECTS OF ORGANIZATION

IMMATERIAL
PRODUCTION

devising
deciding
judging

remembering
recalling
deriving

facts

archiving
providing
transforming

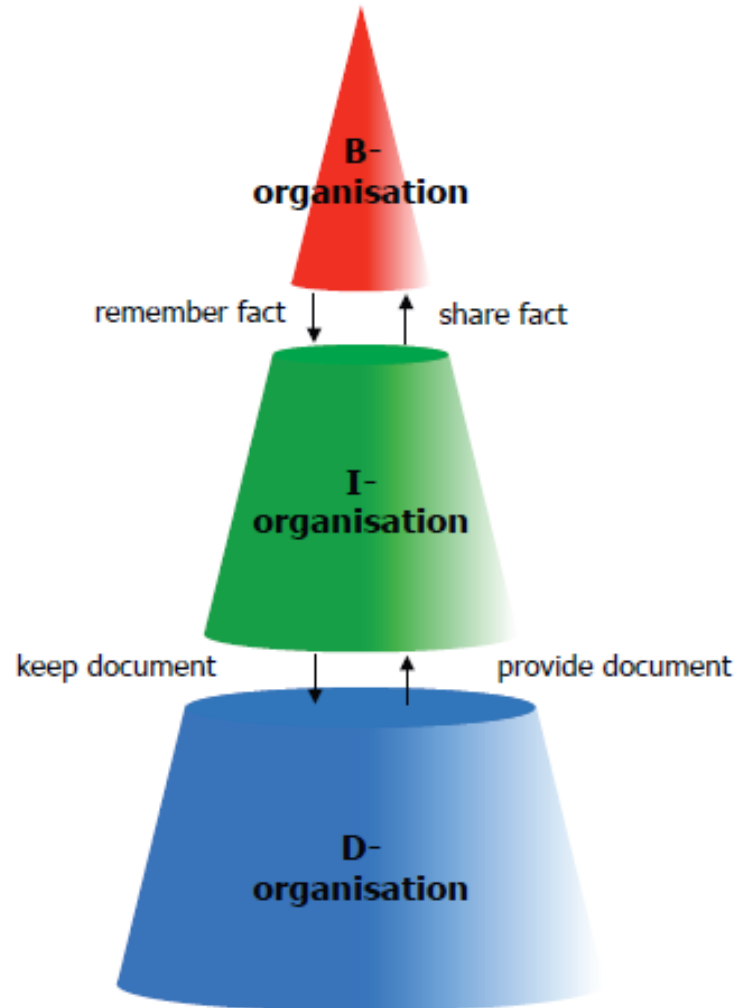
documents

MATERIAL
PRODUCTION

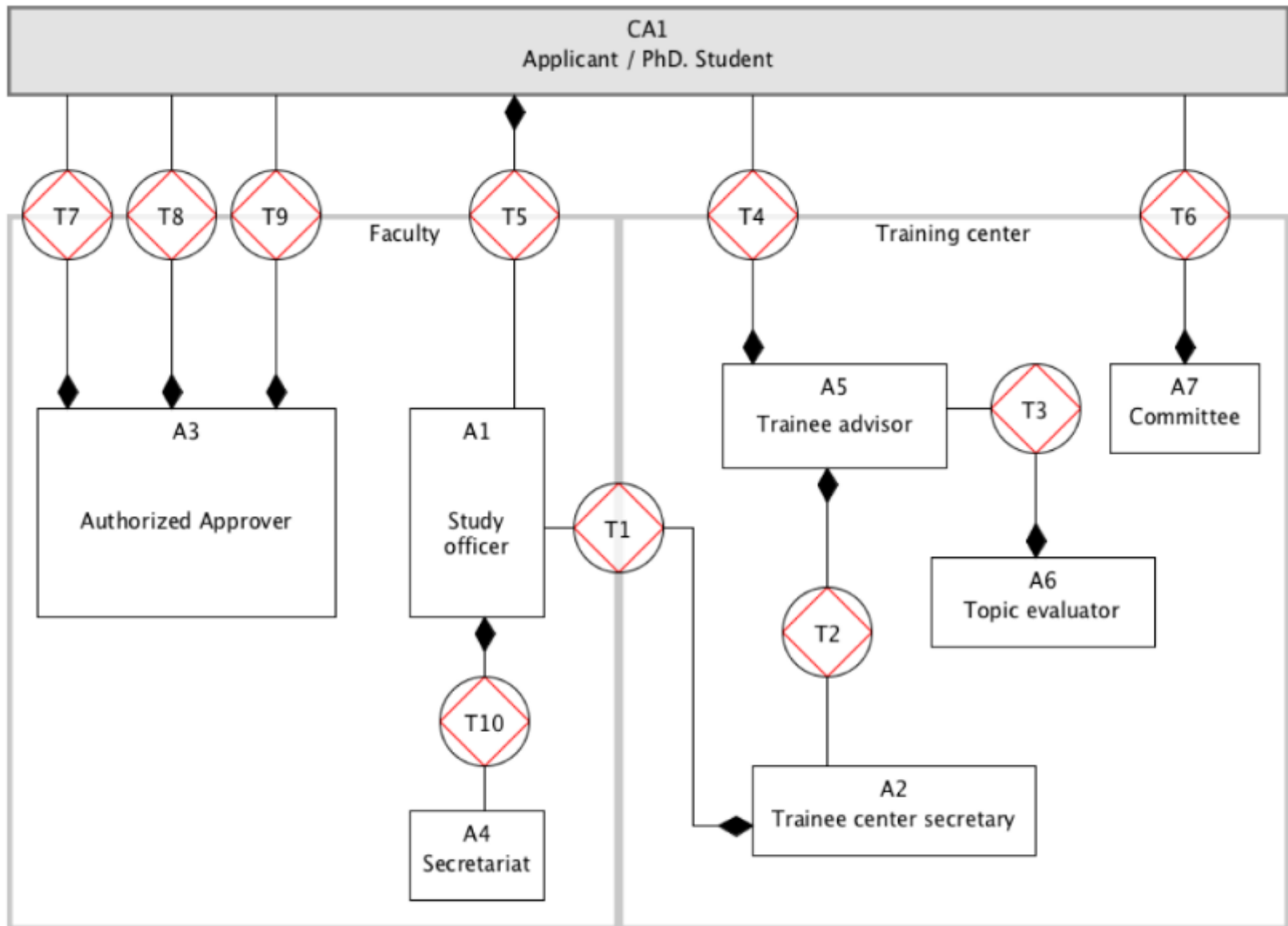
manufacturing
transporting
observing

storing
retrieving
copying
transmitting
destroying

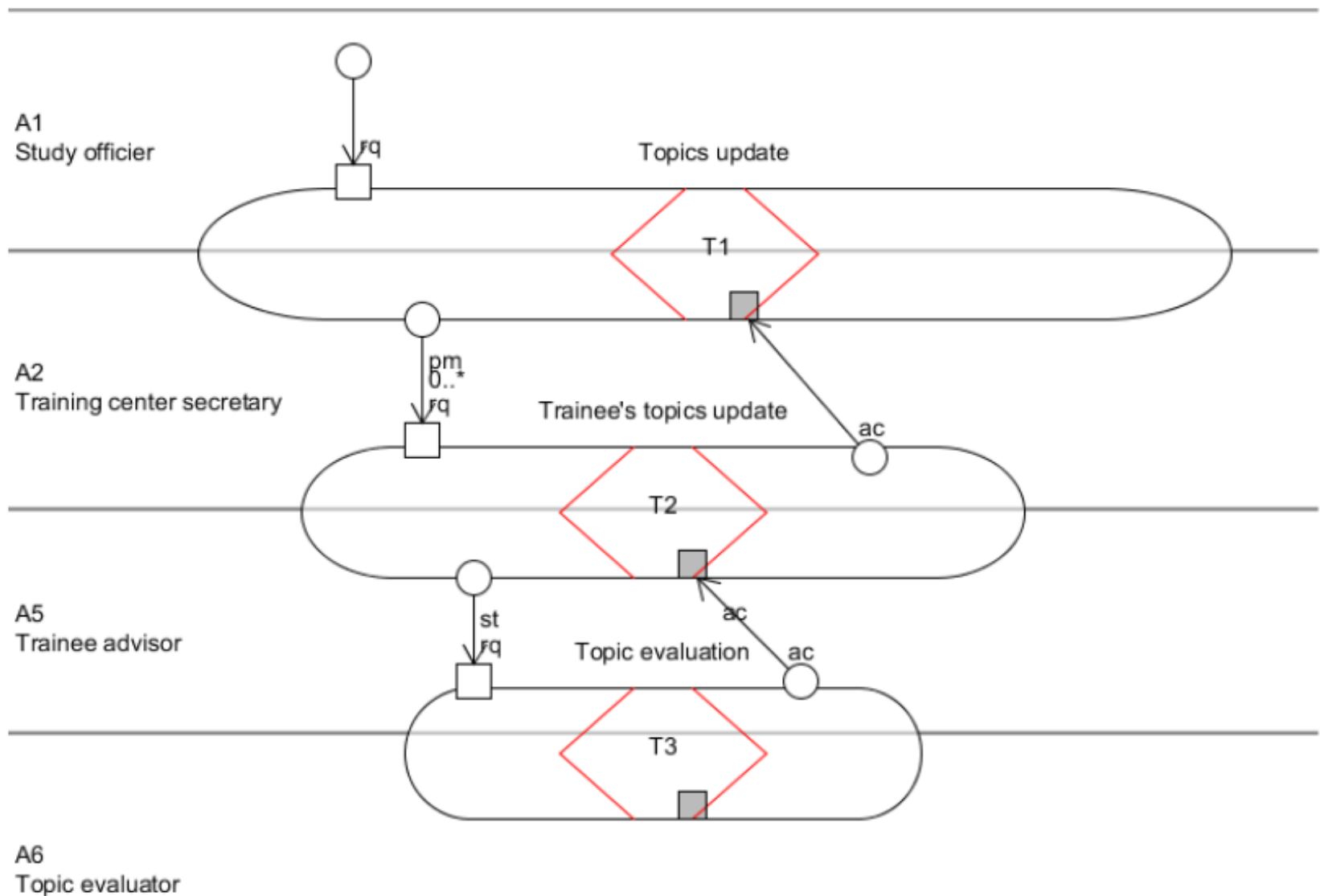
files



OCD DIAGRAM EXAMPLE



PSD DIAGRAM EXAMPLE



ACTOR – ACTOR ROLE MAPPING TABLE

O R G A N I Z A T I O N	ACTOR ROLE									
	01	02	03	04	05	06	07	08	09	10
administrative department										A
application controllers		A	A	A		A				
applicant/PhD. Student					A					
authorized approver							A	A	A	
trainee center secretary	A									

administrative department - study officer, secretariat

application controllers - trainee advisor, topic evaluator, committee





Závěr

Dotazy, připomínky, názory...

Doporučená literatura

- Vymětal, D.: Informační systémy v podnicích – teorie a praxe projektování. ISBN 978-80-247-3046-2. Praha, Grada Publishing 2009
- Keith Robinson, Graham Berrisford: Object Oriented SSADM. Prentice Hall PTR; 1st ed. 1994, ISBN-13: 978-0133094442
- Noran, S.O.: The Fruit and Vegetable Virtual Enterprise: A Virtual Case Study. Griffith University, 1999. Available: <http://www.cit.griffith.edu.au/~noran/>
- Chadima, Z.: Normy a standardy modelování podnikových procesů. 2006. Available: <http://opensoul.iquest.cz/>
- Řepa, V.: Procesní řízení a modelování, 2. rozšířené vydání, ISBN 978-80-247-2252-8. Praha, Grada Publishing 2007
- Dokumenty IDEF0, IDEF1, IDEF1X společnosti Knowledge Based Systems, Inc. (KBSI) : <http://www.idef.com>
- Ashworth, C. and Goodland, M.: SSADM. A Practical Approach, McGraw-Hill, 1992
- Ráček, Jaroslav: Strukturovaná analýza systémů, Brno: Masarykova univerzita, 2006. FI. ISBN 80-210-4190-0
- Jan L. G. Dietz, Enterprise Ontology, Theory and Methodology, Springer-Verlag Berlin Heidelberg, 2006