5t - 14.3.2016 – pondelok

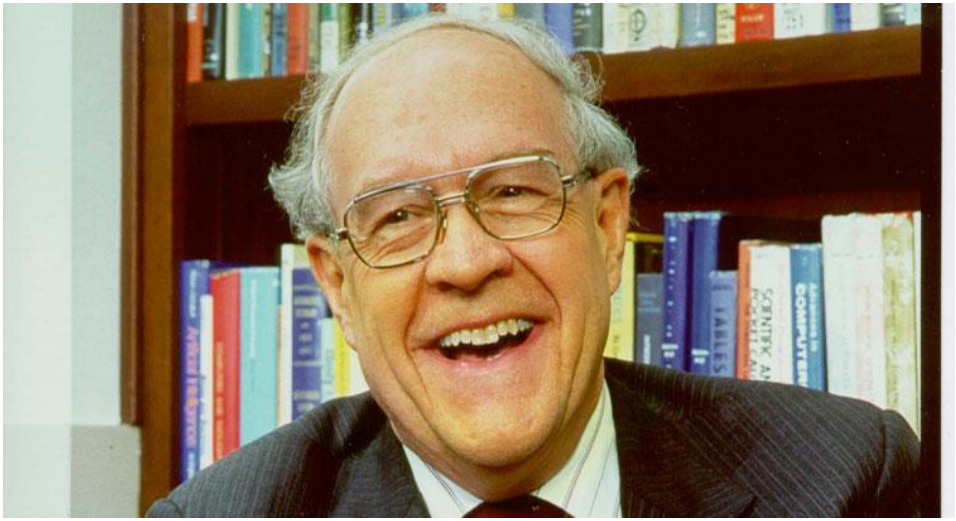
- ✓ tutorial k modelovaniu prípadov použitia v EA
- ✓ opakovanie a otázky k domácemu štúdiu
  - dimenzie modelovania
  - testom riadený vývoj
  - vlastnosti požiadaviek
  - body prípadov použitia
- ✓ softvérové procesy
- ✓ model prípadov použitia
- ✓ model údajov
- **úlohy:**
  - študovať...

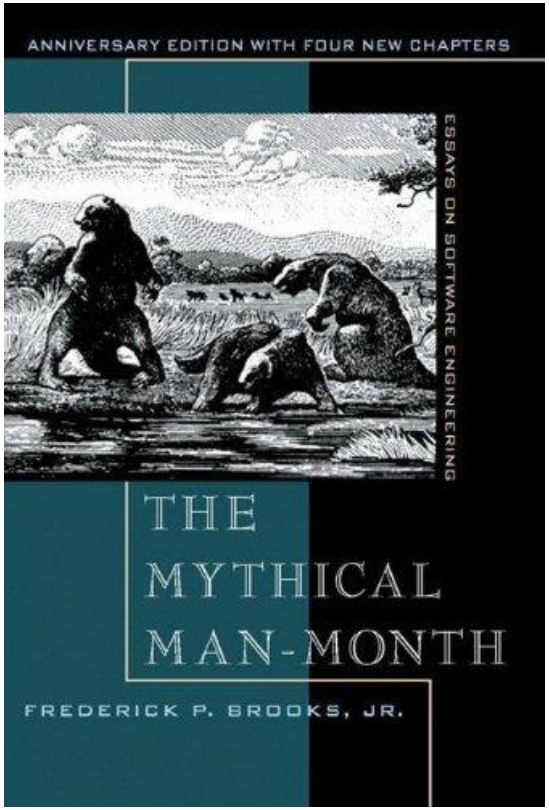
16.3.2016 - Marián Šimko: Modelovanie prípadov použitia a Enterprise Architect (tutoriál)

25.3.2016 - Michal Barla: Manažment verzií

8.4.2016 - Jakub Šimko: Architektúry, návrhové vzory a webové rámce v tvorbe webových aplikácií

22.4.2016 - Ján Suchal: Vybrané témy z implementácie webových aplikácií




ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.

Vnútorné vlastnosti softvéru

- Fred Brooks: No Silver Bullet — Essence and Accidents of Software Engineering. 1986.
  Proceedings of the IFIP Tenth World Computing Conference: 1069–1076.
  IEEE Computer 20 (4): 10–19. 1987

- zložitosť (angl. complexity)
- podriadenosť okoliu (angl. conformity)
- náchylnosť na zmenu, nestálosť, premenlivosť (angl. changeability)
- neviditeľnosť, neuchopiteľnosť (angl. invisibility)

Testovanie

- Postačuje správnosť?
- Je správnosť nevyhnutná?
- Akceptačné testovanie – čierna vs. biela skrinka

UML

Elements of UML 2.0

- ***58. Begin Use-Case Names with a Strong Verb***
- ***59. Name Use Cases Using Domain Terminology***
- ***60. Imply Timing Considerations by Stacking Use Cases***
- ***61. Place Your Primary Actor(s) in the Top Left Corner of the Diagram***
- ***63. Name Actors with Singular, Domain-Relevant Nouns***
- ***65. Name Actors to Model Roles, Not Job Titles***
- ***66. Use*** <<system>> ***to Indicate System Actors***
- ***67. Don't Allow Actors to Interact with One Another***
- ***68. Introduce an Actor Called "Time" to Initiate Scheduled Events***
- ***70. Avoid Arrowheads on Actor–Use-Case Relationships***

Elements of UML 2.0

- ***71. Apply*** <<include>> ***When You Know Exactly When to Invoke the Use Case***
- ***72. Apply*** <<extend>> ***When a Use Case May Be Invoked Across Several Use Case Steps***
- ***74. Generalize Use Cases When a Single Condition Results in Significantly New Business Logic***
- ***76. Avoid More Than Two Levels of Use-Case Associations***
- ***77. Place an Included Use Case to the Right of the Invoking Use Case***
- ***78. Place an Extending Use Case Below the Parent Use Case***

24. Indicate Unknowns with a Question Mark

While you are modeling, you may discover that you do not have complete information. This is particularly true when you are analyzing the domain. You should always try to track down a sufficient answer, but if you cannot do so immediately, then make a good guess and indicate your uncertainty.

Figure 5 depicts a common way to do so with its use of question marks. First, there is a UML note attached to the association between *Professor* and *Seminar* questioning the multiplicity. Second, there is a question mark above the constraint on the *wait listed* association between *Student* and *Seminar*, likely an indication that the modeler isn't sure that it really is a first in, first out (FIFO) list.

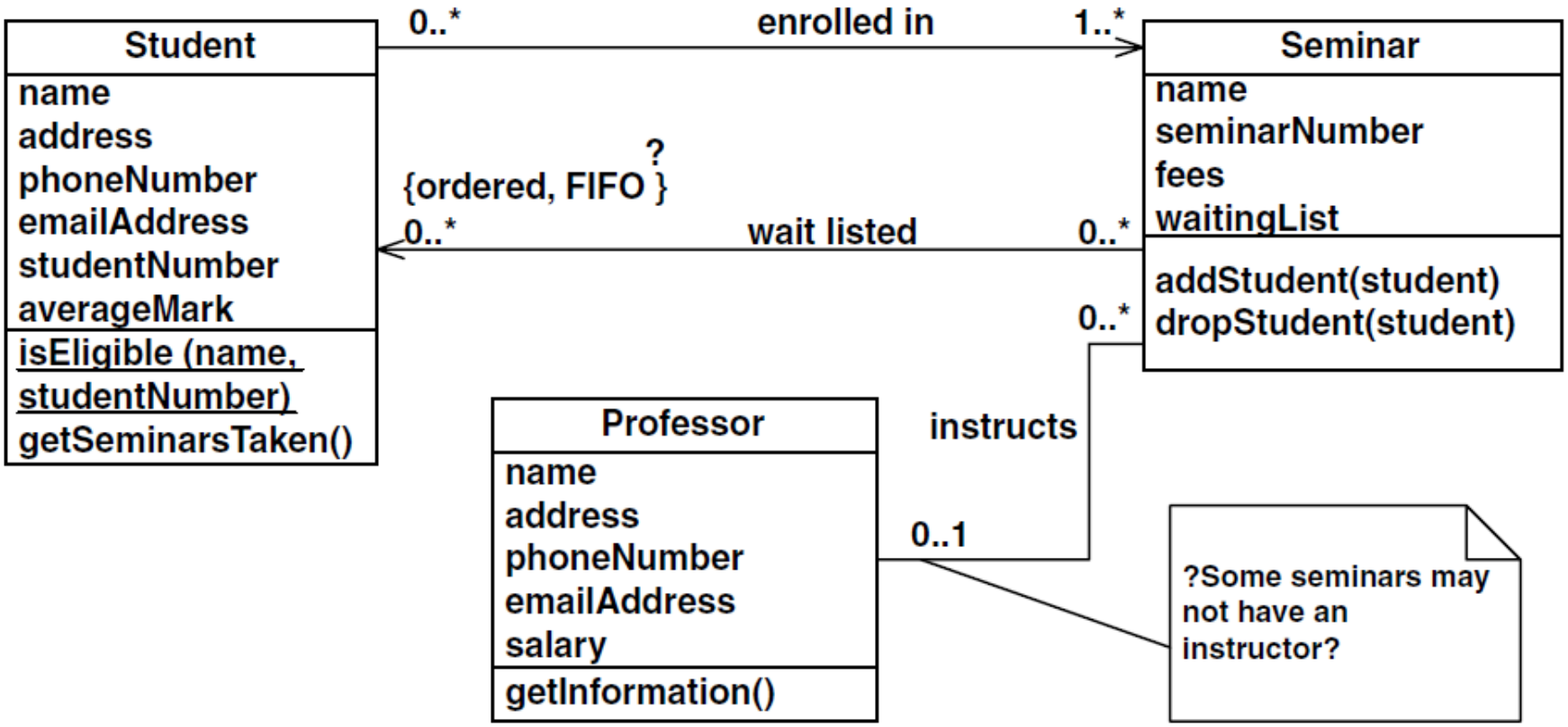


Figure 5. Indicating uncertainty on a diagram.

Model údajov

Úrovne abstrakcie modelov

- Vysoko-úrovňové (konceptuálne) modely
  - Koncepty tak, ako ich vnímajú ľudia
- Prezentačné (implementačné) modely
  - Ľudia ešte dokážu pochopiť
  - Dajú sa jednoducho implementovať v počítači
- Nízko-úrovňové modely (fyzické) modely
  - Podrobnosti o tom, ako sú dáta uložené v počítači

Prednášky DBS, Dr. Michal Barla

Úrovne abstrakcie modelov

- Vysoko-úrovňové (konceptuálne) modely
  - Entitno-relačný model + ďalšie features
- Prezentačné (implementačné) modely
  - Relačný dátový model

- Nízko-úrovňové modely (fyzické) modely
  - Ehm...who cares? :) (o tomto teraz nie)

Prednášky DBS, Dr. Michal Barla

Pravidlá modelovania

- Entity – navzájom odlíšiteľné (majúce identitu) objekty reálneho sveta s nezávislou existenciou
- Vzťahy medzi entitami
  - Majú kardinalitu
- Iba atomické atribúty
  - Čiže žiadne štruktúry, zoznamy hodnôt
    - Hoci to nie je v dnešnom svete úplne pravda
- Snažíme sa zbytočne neduplikovať údaje
- Myslíme na vývoj dát v čase

Prednášky DBS, Dr. Michal Barla

E-R model vs. E-R diagram

- E-R model – množina entít, vzťahov a atribútov
- E-R diagram – vizualizácia modelu vo zvolenej notácii
- UML class diagram umožňuje zachytiť aj operácie nad objektami
- UML notáciu mám najradšej :)

Základné pojmy

- Databáza je kolekcia **relácií** (alebo **tabuliek**)
- Každá relácia má sadu pomenovaných **atribútov** (**stĺpcov**)
- **Riadky** predstavujú inštancie, ktoré majú **hodnoty** pre jednotlivé atribúty
- Každý atribút má **typ** (**doménu**)

Študenti

ID	meno	všp	fotka
123	Fero	2.1	:-)
854	Eva	3.5	}(:-|
...			
..			

Jedálne

meno	lokalita	kapacita
horná	atriáky	300
dolná	atriáky	150
študentská	FEI	80

Prednášky DBS, Dr. Michal Barla

Základné pojmy

- Schéma – štruktúra tabuliek
- Inštancie – dáta, obsah tabuliek

Študenti

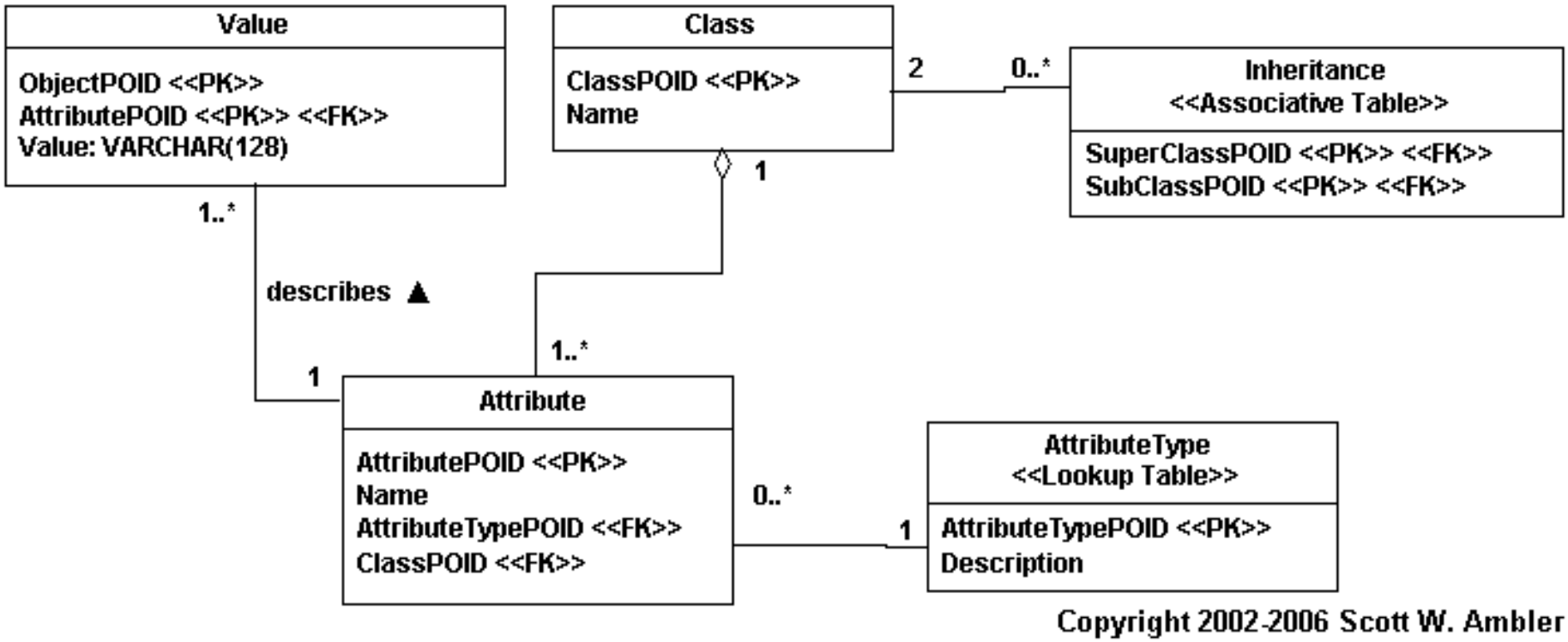
ID	meno	všp	fotka
123	Fero	2.1	:-)
854	Eva	3.5	}(:-|
...			
..			
.			

Jedálne

meno	lokalita	kapacita
horná	atriáky	300
dolná	atriáky	150
študentská	FEI	80

Prednášky DBS, Dr. Michal Barla

Generická schéma dát



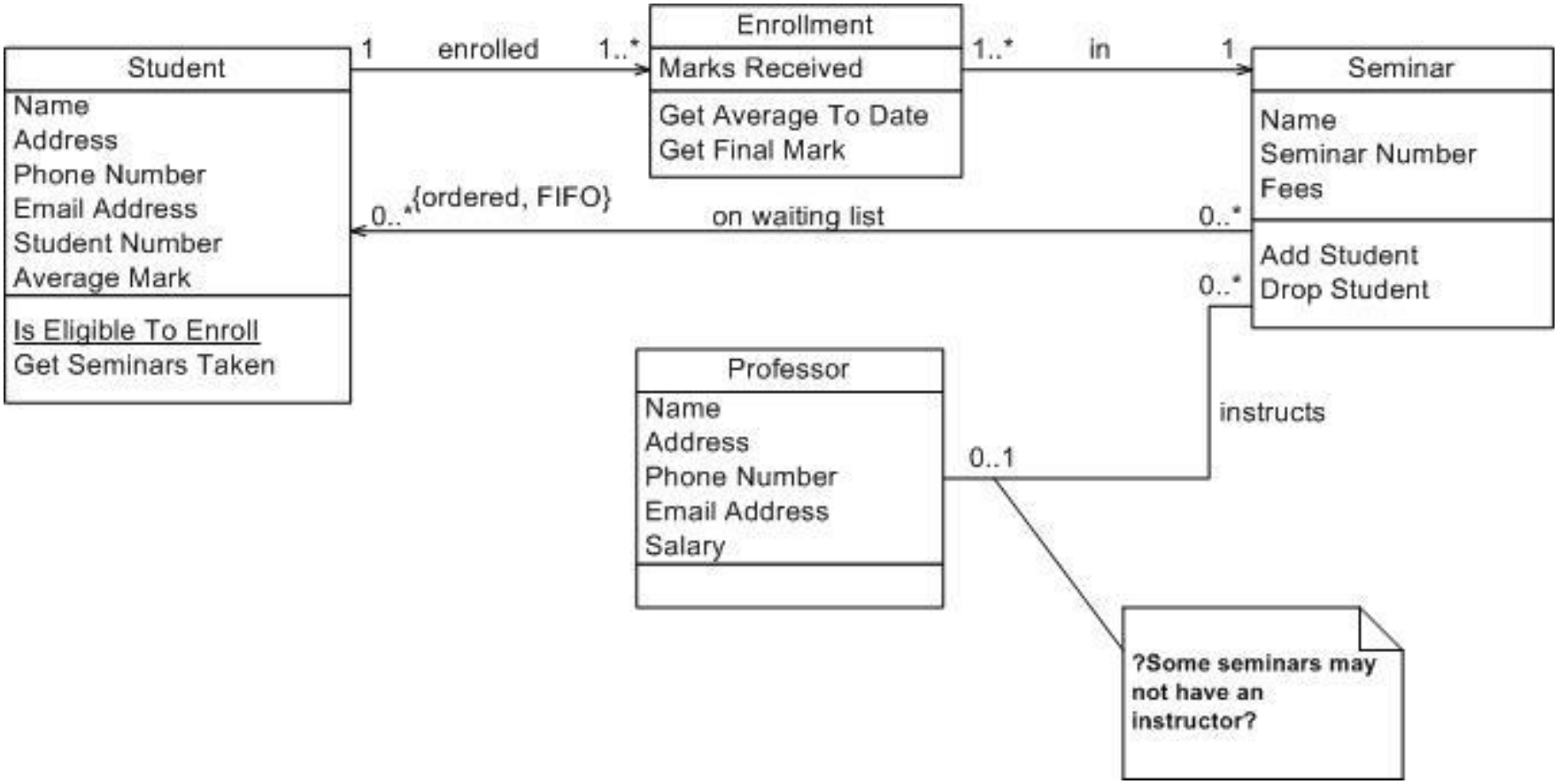
Štruktúrne modely – UML

- Diagram tried
- Diagram objektov

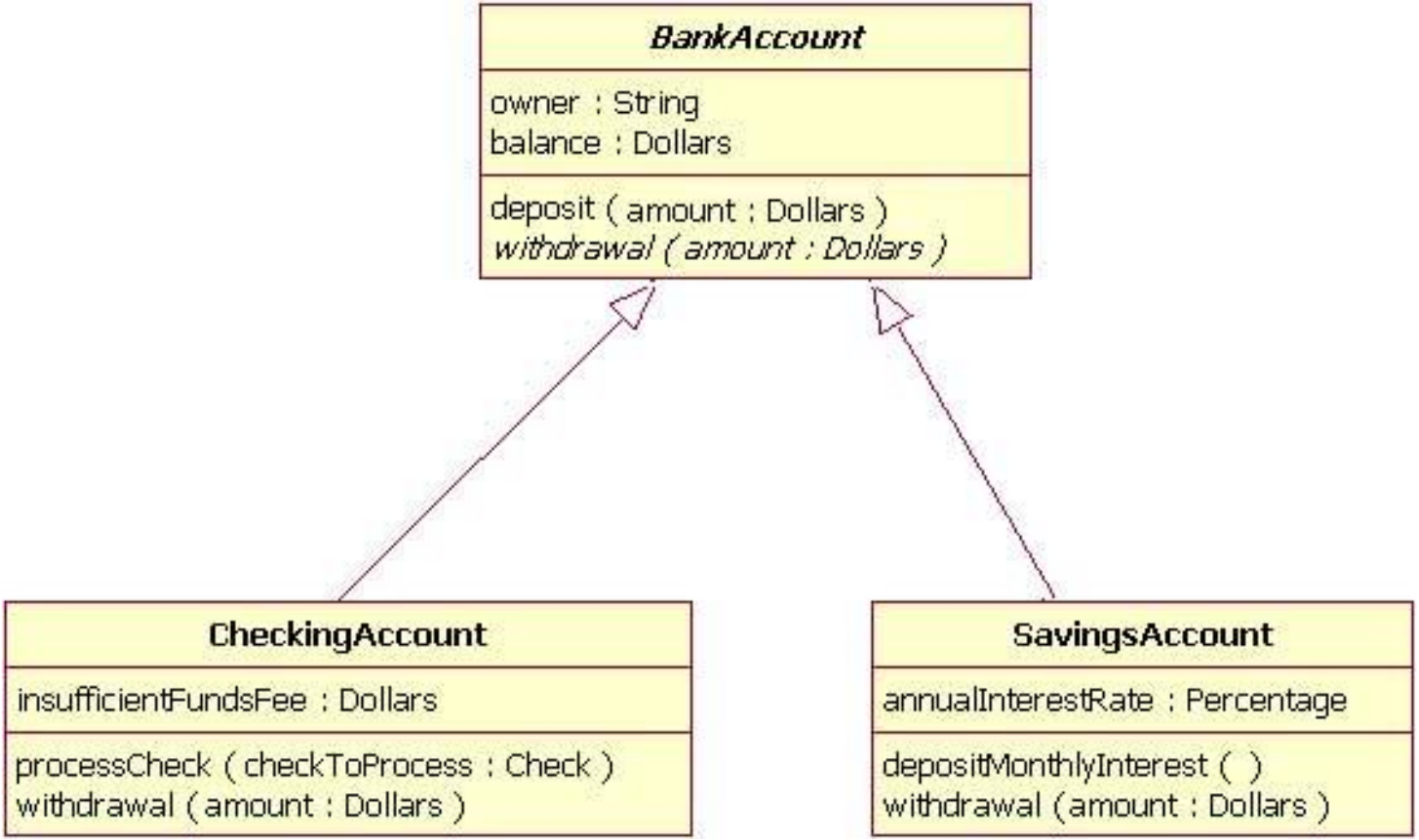
POROVNANIE

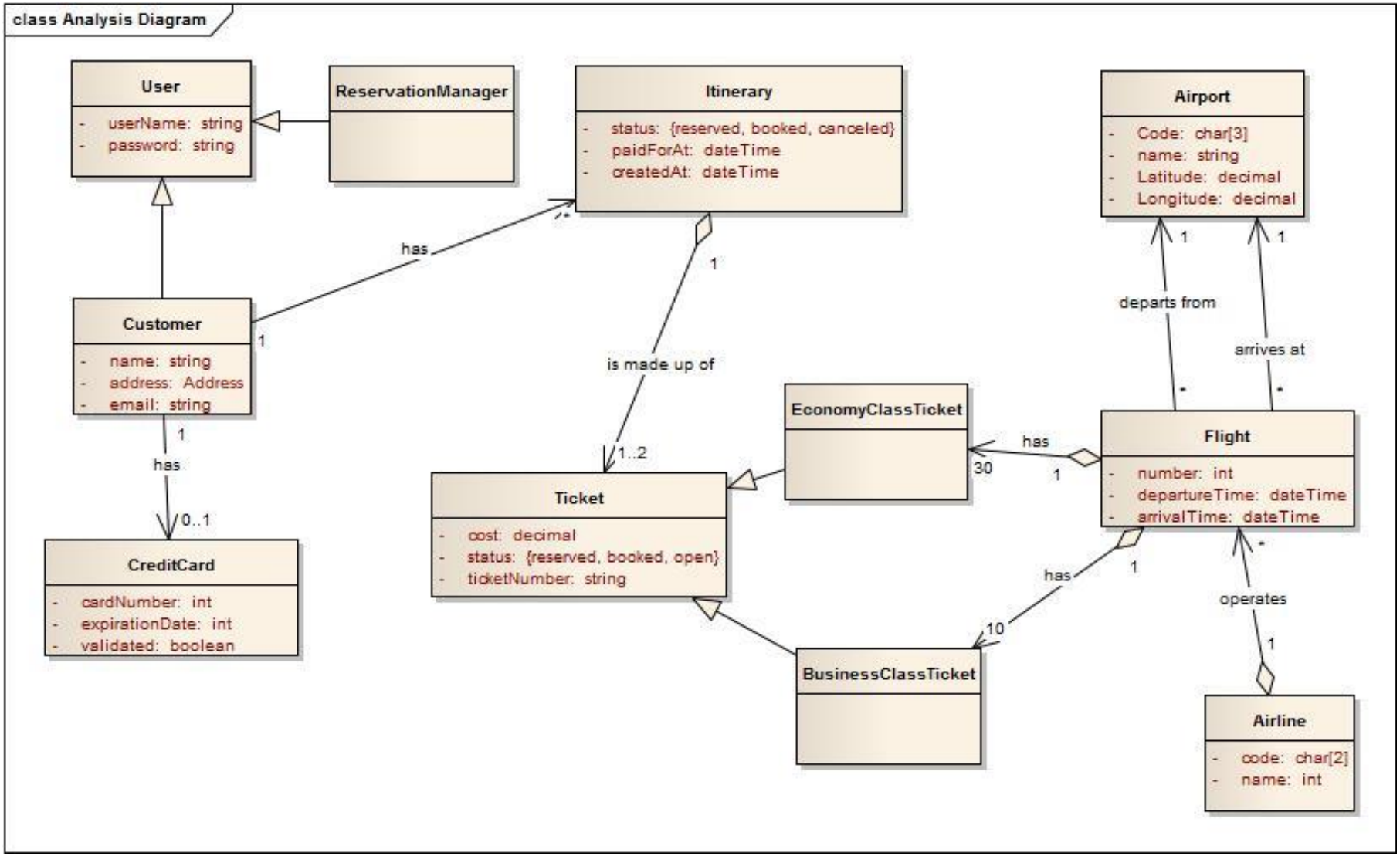
- notácia, typ modelu
- použitie
- etapa životného cyklu

Diagram tried



Generalizácia

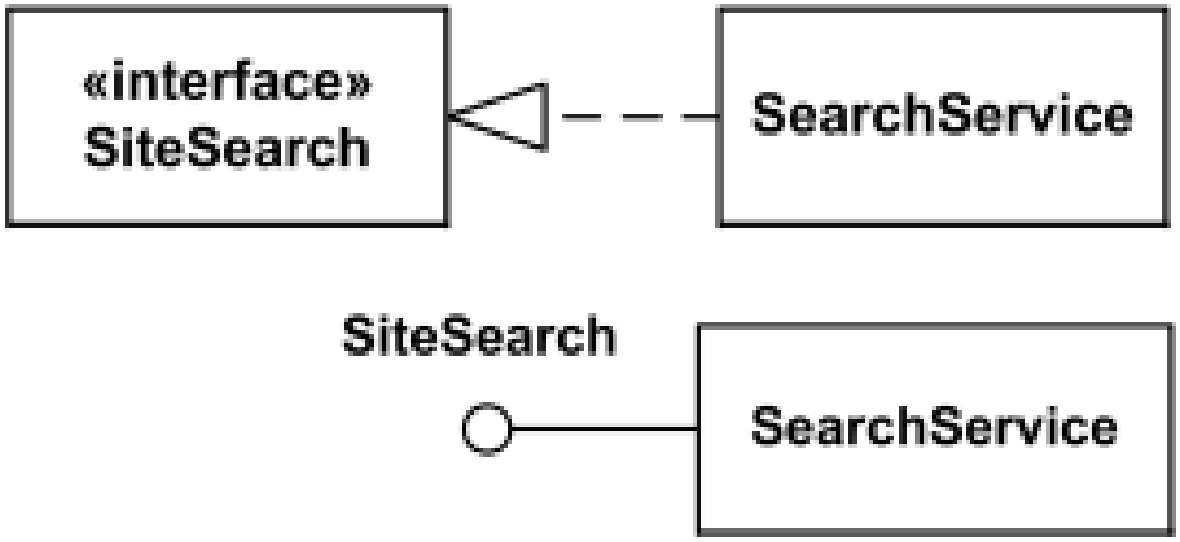



class Analysis Diagram
User
- userName: string
- password: string
ReservationManager
Itinerary
- status: {reserved, booked, canceled}
- paidForAt: dateTime
- createdAt: dateTime
Airport
- Code: char[3]
- name: string
- Latitude: decimal
- Longitude: decimal
Customer
- name: string
- address: Address
- email: string
CreditCard
- cardNumber: int
- expirationDate: int
- validated: boolean
Ticket
- cost: decimal
- status: {reserved, booked, open}
- ticketNumber: string
EconomyClassTicket
BusinessClassTicket
Flight
- number: int
- departureTime: dateTime
- arrivalTime: dateTime
Airline
- code: char[2]
- name: int
has
1
*
1
1
1..2
has
0..1
has
30
1
departs from
arrives at
1
1
*
*
has
1
10
operates
*
1

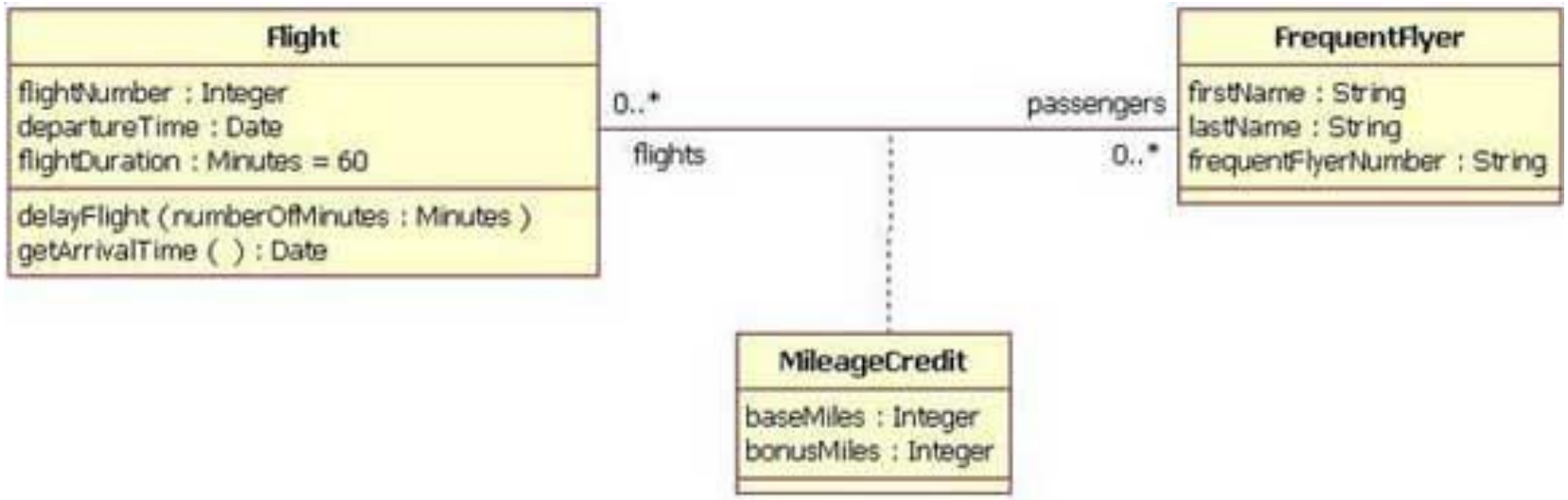
18. Focus on Content First, Appearance Second

There is always the danger of adding hours onto your CASE tool modeling efforts by rearranging the layout of your symbols and lines to improve the diagram's readability. The best approach is to focus on the content of a diagram at first and only try to get it looking good in a rough sort of way—it doesn't have to be perfect while you're working on it. **Once you're satisfied that your diagram is accurate enough, and that you want to keep it, then invest the appropriate time to make it look good.** An advantage of this approach is that you don't invest significant effort improving diagrams that you eventually discard.

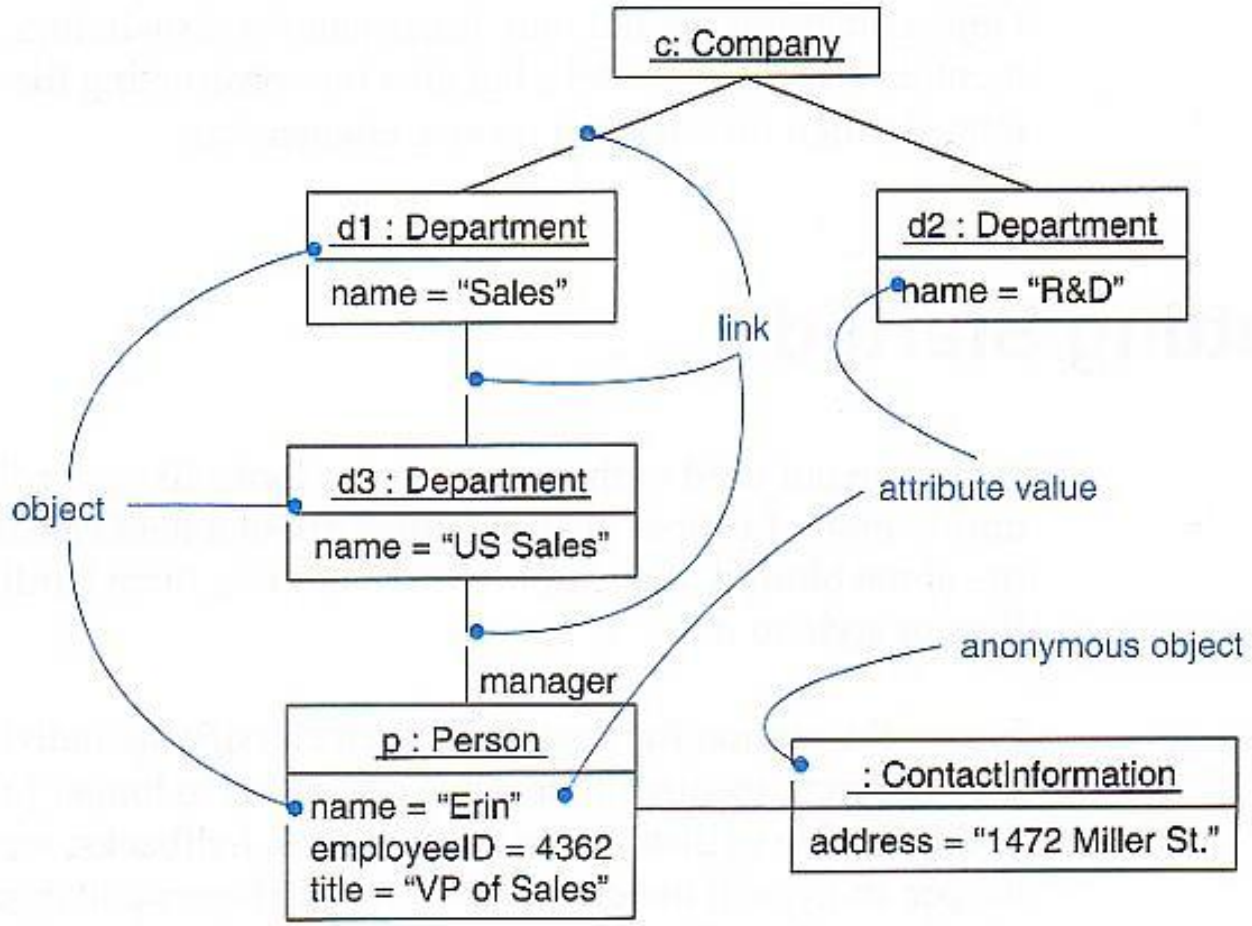
Realizácia rozhrania

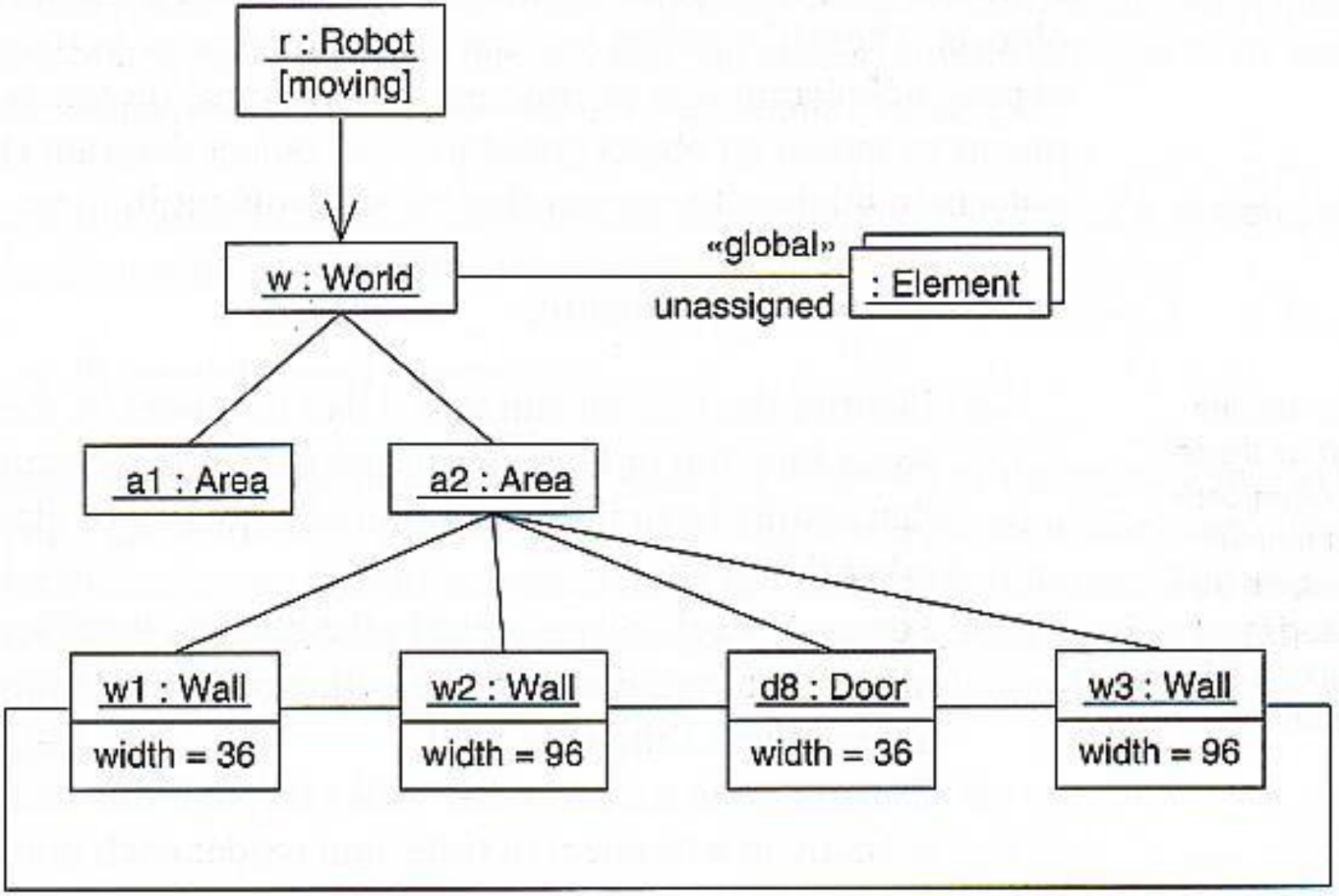


Väzobná entita



Objektový diagram




r : Robot
[moving]
w : World
«global»
unassigned
: Element
a1 : Area
a2 : Area
w1 : Wall
width = 36
w2 : Wall
width = 96
d8 : Door
width = 36
w3 : Wall
width = 96

Enroll In Seminar
Basic Course of Action

1. Student indicates wish to enroll

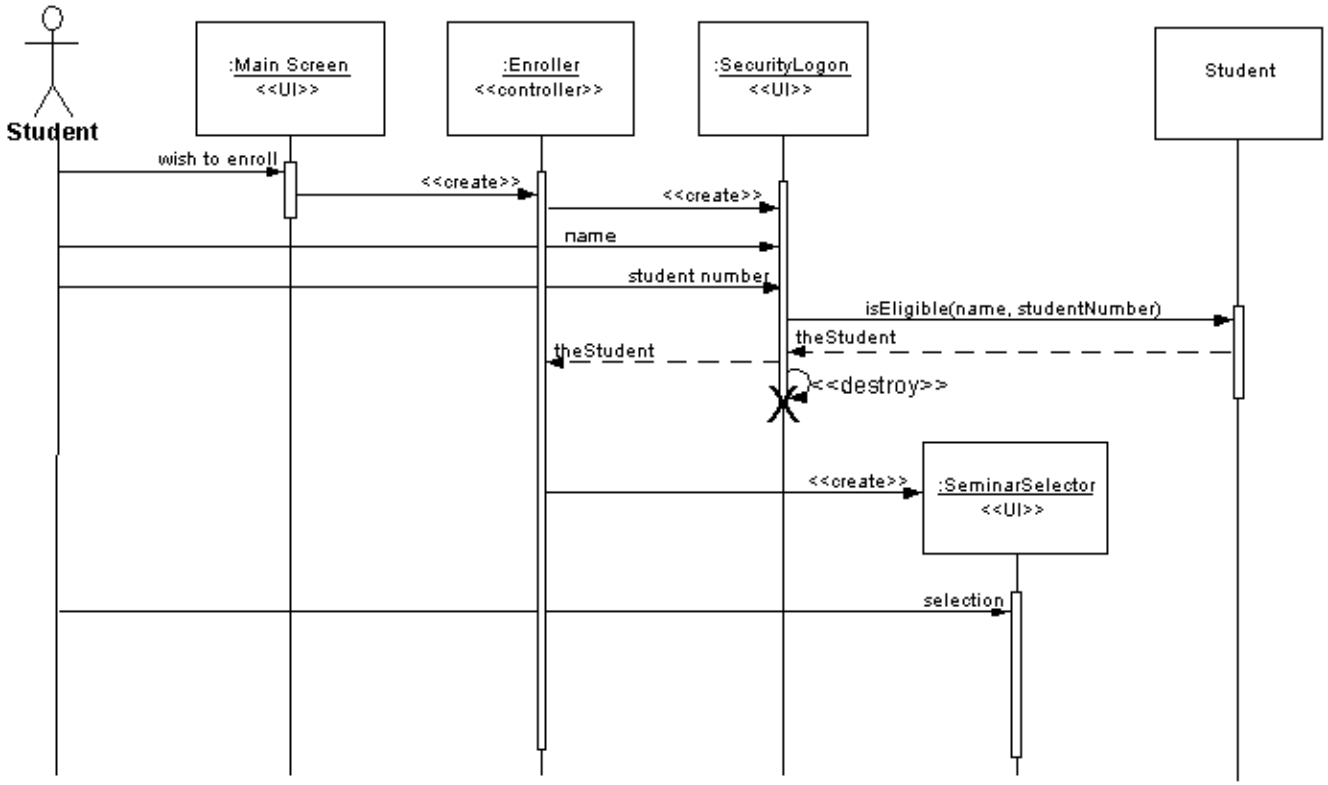
2. Student inputs name and number

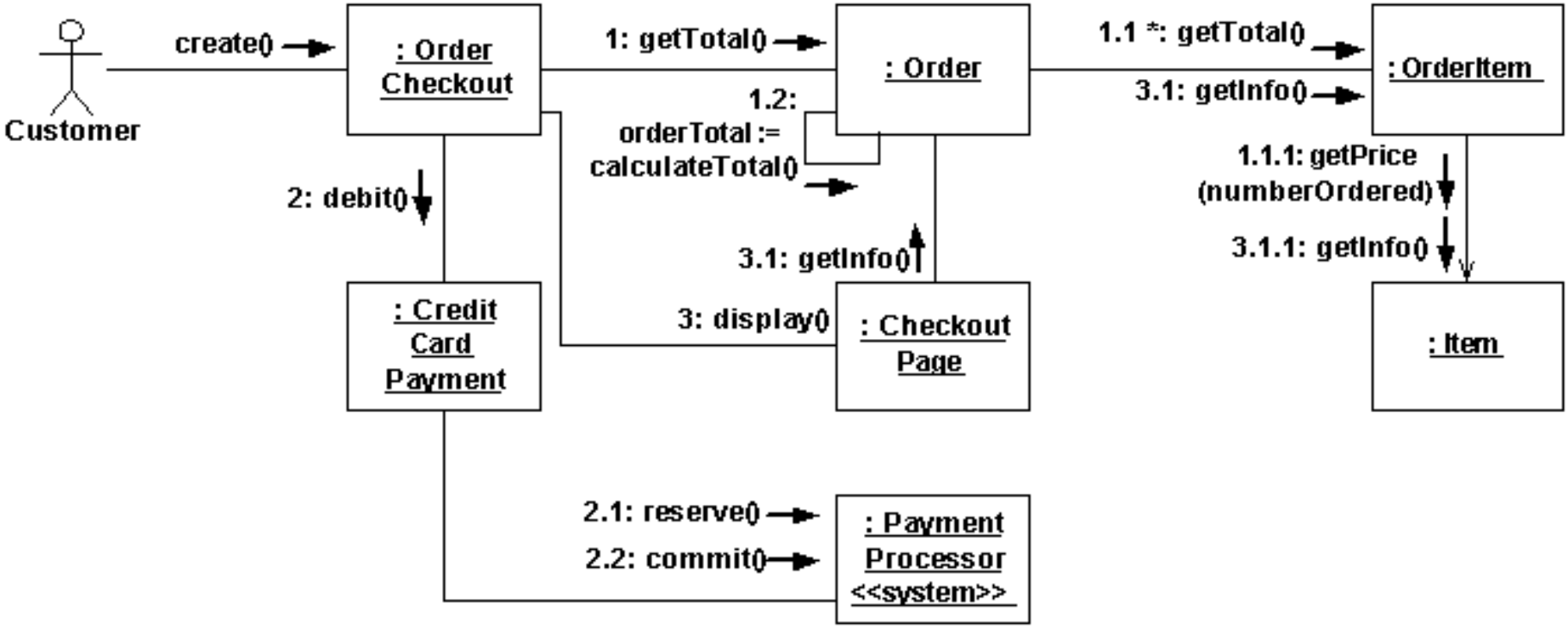
3. System verifies student

4. System displays seminar list

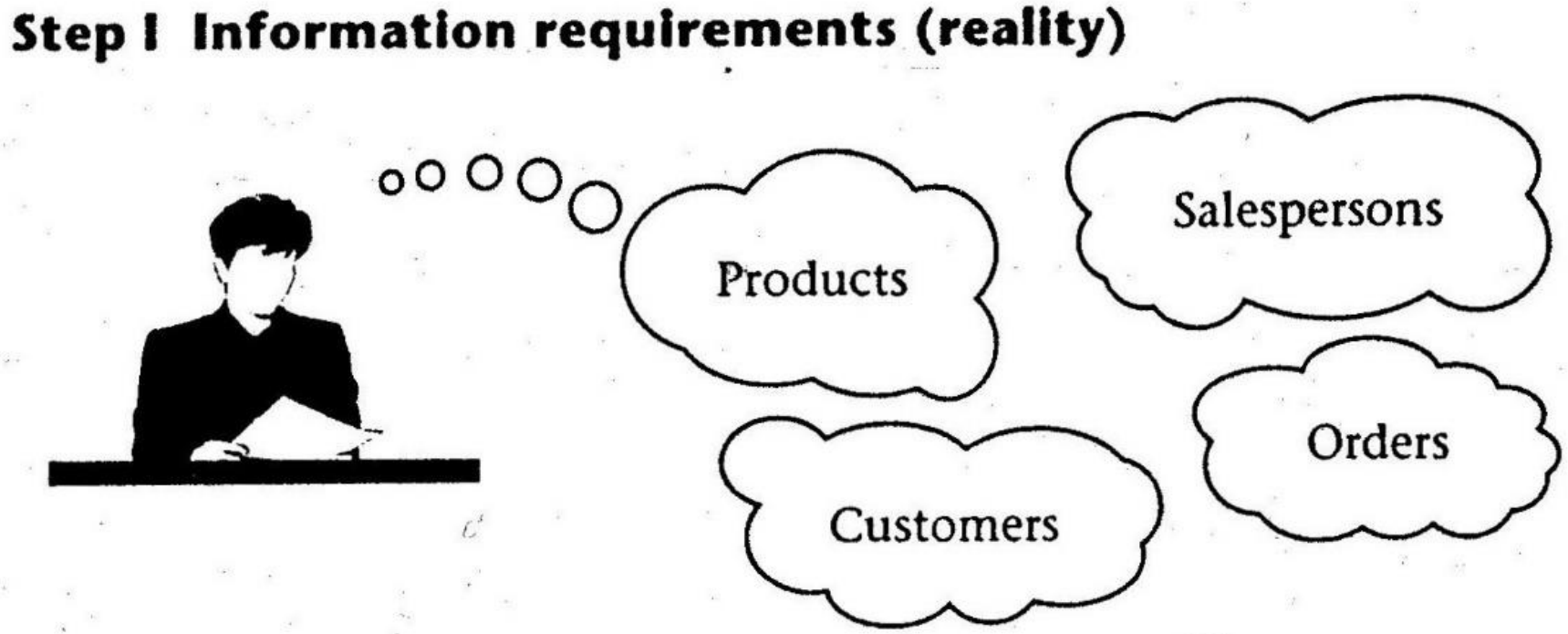
5. Students picks seminar

...




Customer
create()
: Order Checkout
1: getTotal()
: Order
1.1 *: getTotal()
: OrderItem
3.1: getInfo()
1.2:
orderTotal :=
calculateTotal()
1.1.1: getPrice
(numberOrdered)
3.1.1: getInfo()
2: debit()
3.1: getInfo()
: Credit Card Payment
3: display()
: Checkout Page
: Item
2.1: reserve()
2.2: commit()
: Payment Processor
<<system>>

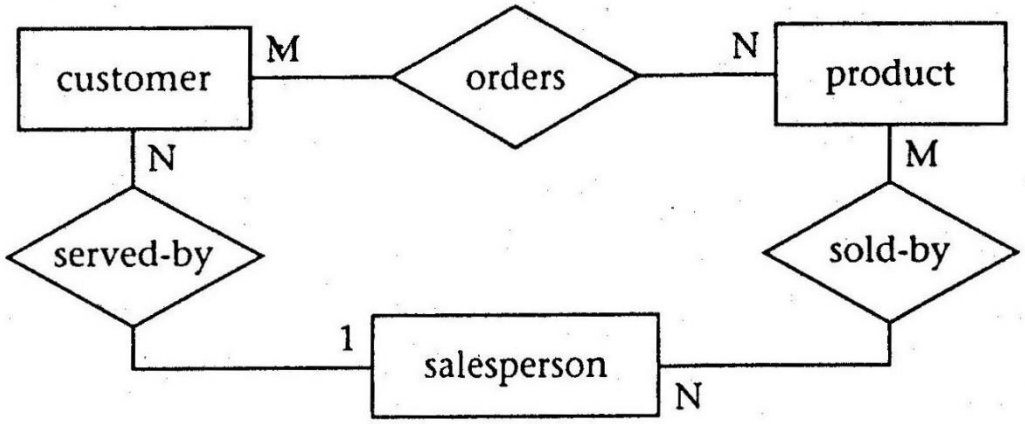
Postup tvorby modelu údajov



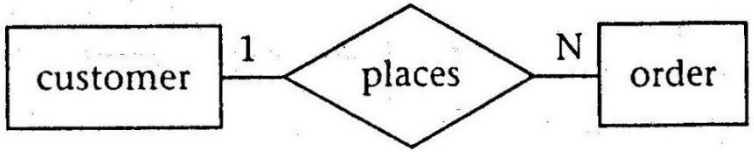
Step II Logical design

Step II(a) ER modeling (conceptual)

Retail salesperson view

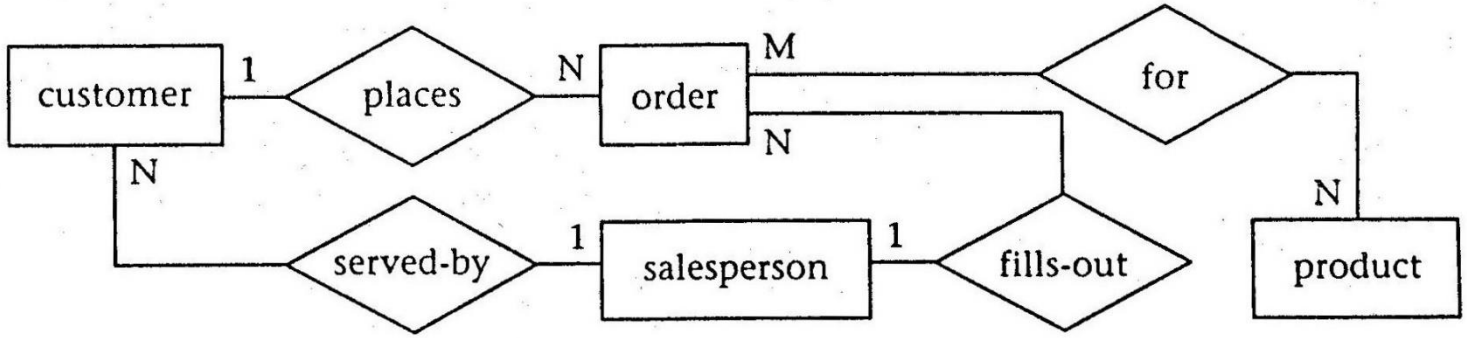


Customer view



Step II(b) View integration

Integration of retail salesperson's and customer's views



Step II(c) Transformation of the ER diagram to SQL tables

Customer

cust-no	cust-name	. . .

Product

prod-no	prod-name	qty-in-stock

Salesperson

sales-name	addr	dept	job-level	vacation-days

Order

order-no	sales-name	cust-no

Order-product

order-no	prod-no

```
create table customer
    (cust_no integer,
    cust_name char(15),
    cust_addr char(30),
    sales_name char(15),
    prod_no integer,
    primary key (cust_no),
    foreign key (sales_name)
        references salesperson,
    foreign key (prod_no)
        references product);
```

Step II(d) Normalization of SQL tables (3NF, BCNF, 4NF, 5NF)

Decomposition of tables and removal of update anomalies

Salesperson

sales-name	addr	dept	job-level

Sales-vacations

job-level	vacation-days

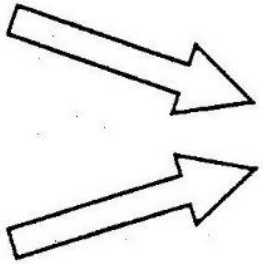
Step III Physical design (including denormalization)

Customer

cust-no	cust-name

Order

order-no	sales-name	cust-no

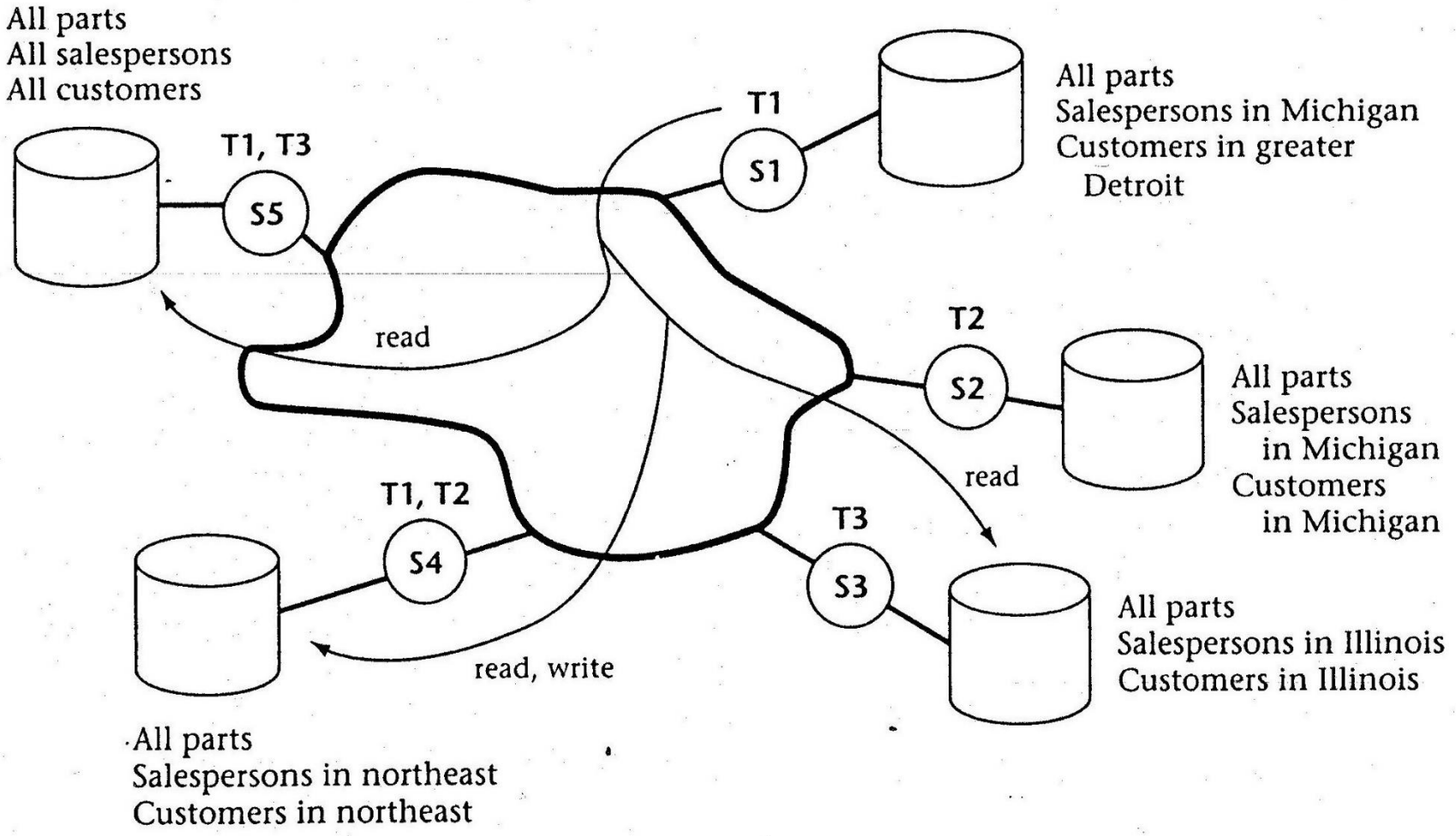


Customer / refined

cust-no	cust-name	sales-name

Physical design parameters:
indexing, access methods, clustering

Step IV Data distribution



S1 = Ann Arbor, S2 = Detroit, S3 = Chicago, S4 = Boston, S5 = New York

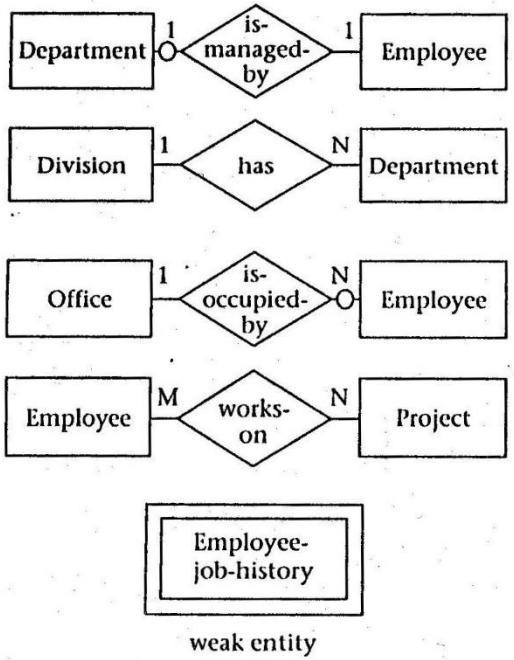
T1, T2, T3 are transactions (the figure shows all sites where they are initiated)

Decisions: fragmentation, replication, allocation

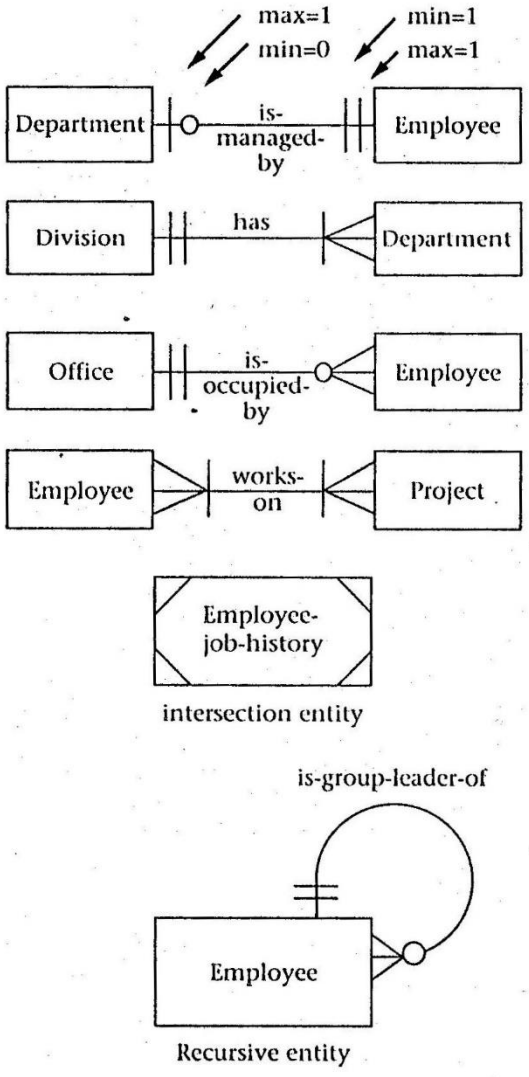
Objectives: minimum response time, minimum communication cost, maximum availability

Comparison of ER Construct Conventions

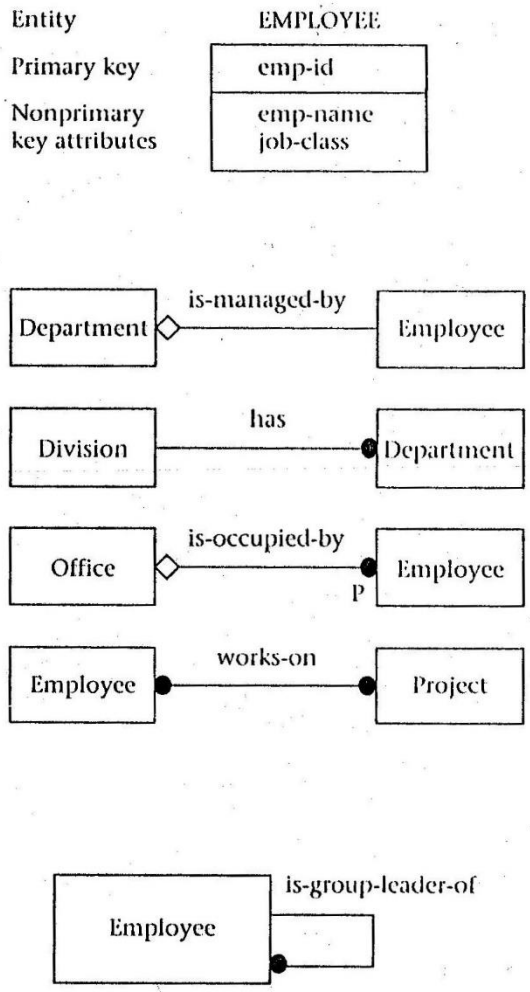
ER model constructs using the Chen notation

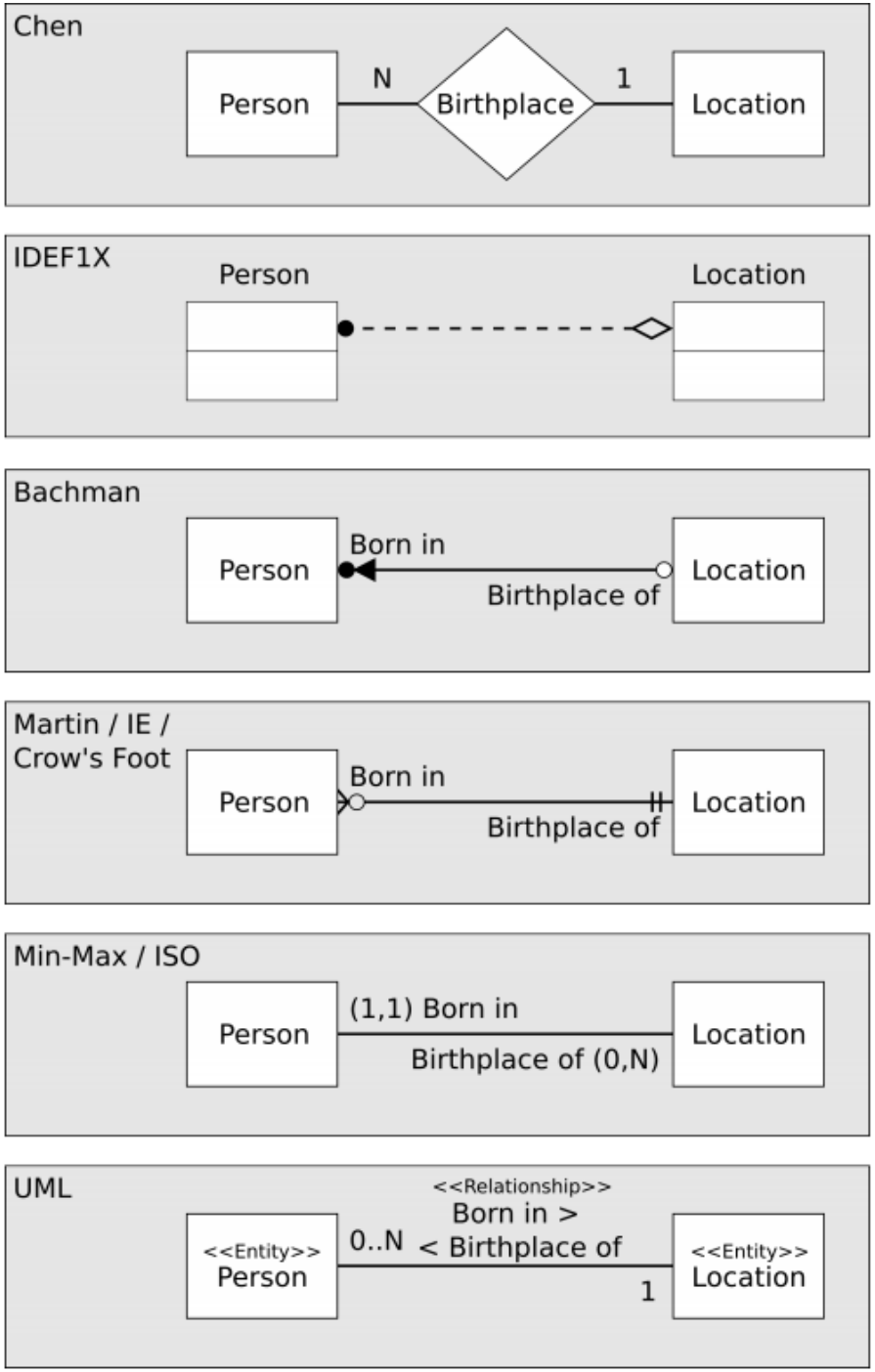


ER model constructs using the "crow's foot" approach [Ever86, Knowledgeware]

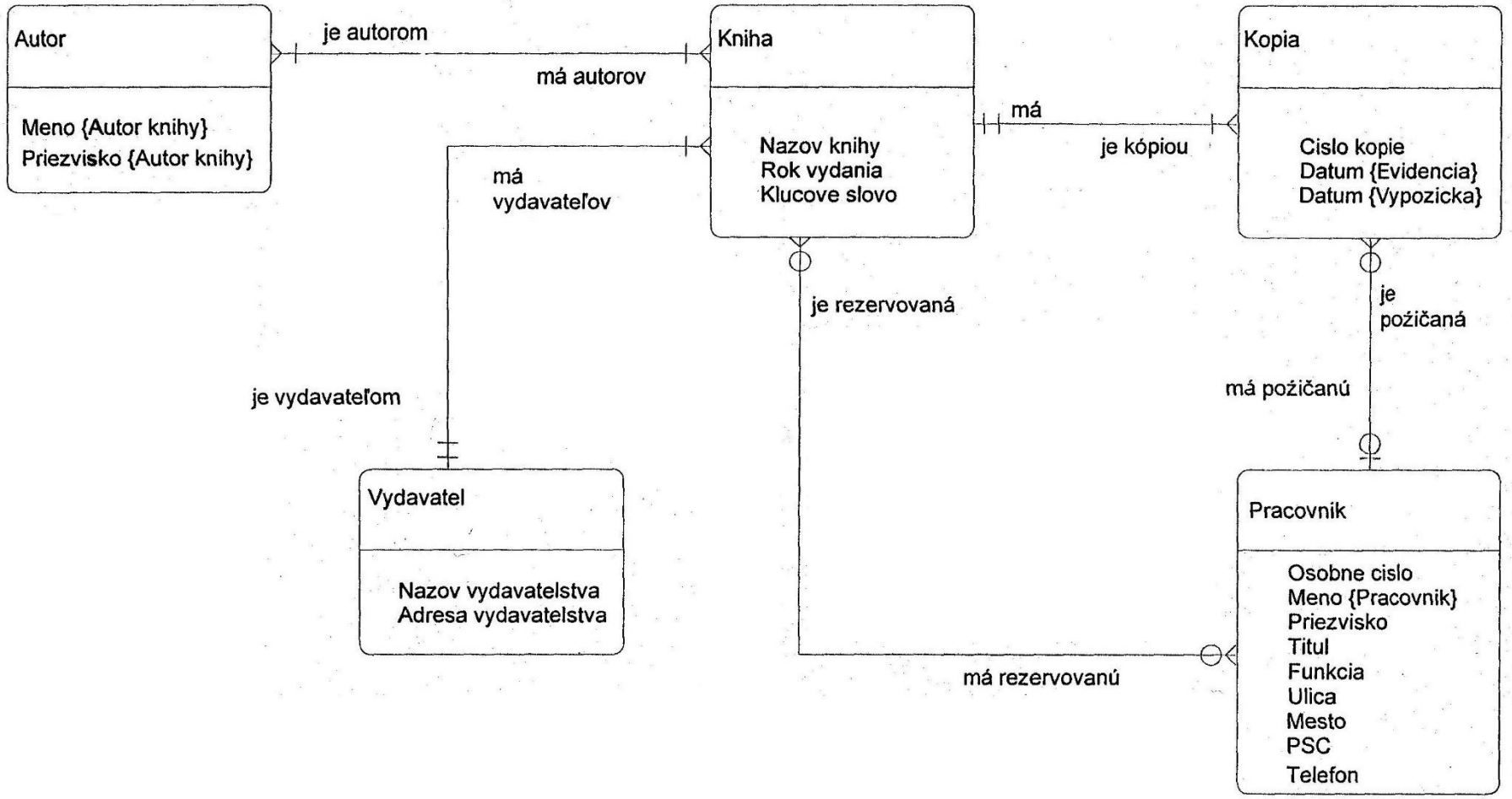


ER model constructs using IDEF1X [Bruc92]

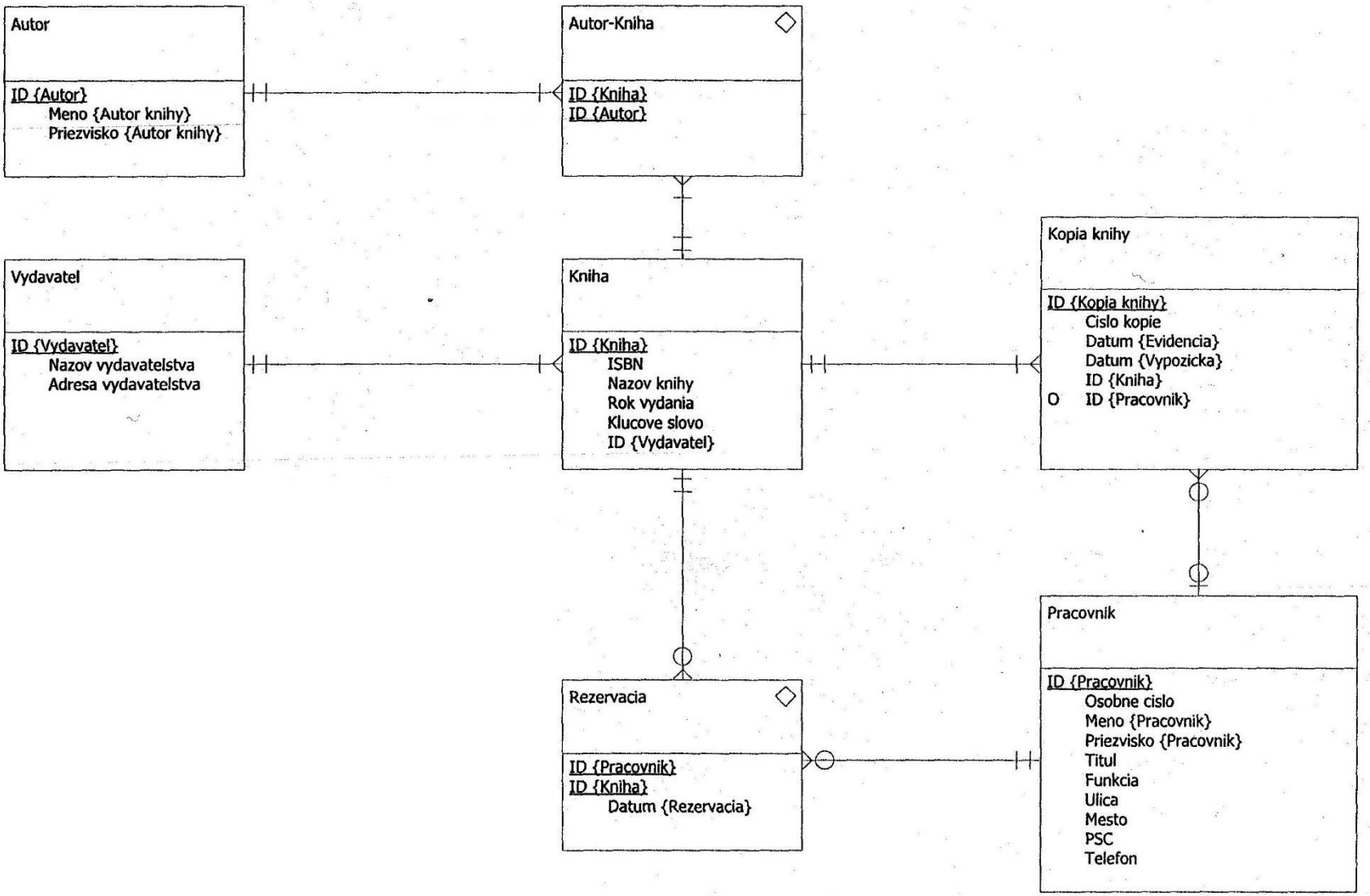



Chen
Person
N
Birthplace
1
Location
IDEF1X
Person
Location
Bachman
Person
Born in
Birthplace of
Location
Martin / IE /
Crow's Foot
Person
Born in
Birthplace of
Location
Min-Max / ISO
Person
(1,1) Born in
Birthplace of (0,N)
Location
UML
<<Relationship>>
Born in >
< Birthplace of
<<Entity>>
Person
0..N
1
<<Entity>>
Location

Logický model údajov



Fyzický model údajov



PRÍKLAD ENTITNO RELAČNÉHO DIAGRAMU (Software through Pictures)