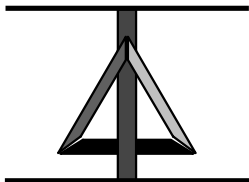


# Data Model Patterns: A Metadata Map

***East Coast DAMA 2006***

David C. Hay

*May-June, 2006*



**Essential Strategies, Inc.**

13 Hilshire Grove Lane, Houston, TX 77055

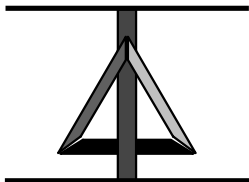
☎ (713) 464-8316

✉ [dch@essentialstrategies.com](mailto:dch@essentialstrategies.com)

🌐 [www.essentialstrategies.com](http://www.essentialstrategies.com)

*There once was a fellow named Corey  
Whose career was not covered in glory  
He had a bad day  
When he just couldn't say  
Metadata Repository.*

## Introduction



### **Essential Strategies, Inc.**





13 Hilshire Grove Lane, Houston, TX 77055

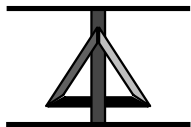
☎ (713) 464-8316

✉ [dch@essentialstrategies.com](mailto:dch@essentialstrategies.com)

🌐 [www.essentialstrategies.com](http://www.essentialstrategies.com)

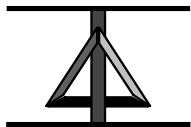
# About Metadata . . .

This Book (Meta-metadata)	<b>Elements of metadata (metadata model)</b>	Objects: "Entity Class", "Attribute"	Objects: "Entity Class" "Attribute" "Role"	Objects: "Table" "Column"	Object: "Program module", "Language"
Data Management (Metadata)	<b>Data about a database (a data model)</b>	Entity class: "Customer" Attributes: "Name" "Birthdate"	Entity classes: "Branch", "Employee" Attributes: "Employee.Address" "Employee.Name" Role: "Each Branch must be managed by exactly one Employee"	Table: "CHECKING_ACCOUNT" Columns: "Account_number" "Monthly_charge"	Program module: ATM Controller Language: Java
IT Operations (Instance Data)	<b>Data about real world things (a database)</b>	Customer Name: "Julia Roberts"; Customer Birthdate: "10/28/67"	Branch Address: "111 Wall Street" Branch Manager: "Sam Sneed"	CHECKING_ACCOUNT. Account_number: = "09743569" CHECKING_ACCOUNT. Monthly_charge: "\$4.50"	ATM Controller: Java code
	<b>Real world things</b>	Julia Roberts 	Wall Street branch 	Checking account #09743569 	ATM Withdrawal 



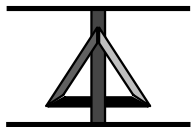
# The Architecture Framework . . .

	<i>Data (What)</i>	<i>Activities (How)</i>	<i>Locations (Where)</i>	<i>People (Who)</i>	<i>Time (When)</i>	<i>Motivation (Why)</i>
<b>Objectives / Scope (Planner's view)</b>	List of things important to the enterprise 	List of processes the enterprise performs 	List of enterprise locations 	Organization approaches 	Business master schedule 	Business vision and mission 
<b>Enterprise model (Business Owners' Views)</b>	Language, divergent data model 	Business process model 	Logistics network 	Organization chart 	State / transition diagram 	Business strategies, tactics, policies, rules 
<b>Model of Fundamental Concepts (Architect's View)</b>	Convergent e/r model 	Essential data flow diagram 	Locations of roles 	The viable system, use cases 	Entity Life History 	Business rule model 
<b>Technology Model (Designer's View)</b>	Data base design 	System design, program structure 	Hardware, software distribution 	User interface, security design 	Control structure 	Business rule design 
<b>Detailed Representation (Builder's View)</b>	Physical storage design 	Detailed program design 	Network architecture, protocols 	Screens, security coding 	Timing definitions 	Rule specification program logic 
<b>Functioning System</b>	<i>(Working System)</i>					
	Converted data	Executable programs	Communications facilities	Trained people	Business events	Enforced rules



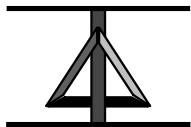
## The Rows . . .

- ***Planner's View:*** Scope, Vision, Mission
- ***Business Owner's View:*** The people who do the work
- ***Architect's View:*** Looking for a single, integrated view
- ***Designer's View:*** Using technology to solve problems
- ***Builder's View:*** The nuts and bolts of programming, database administration
- ***The Functioning System:*** Systems as they exist

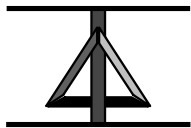
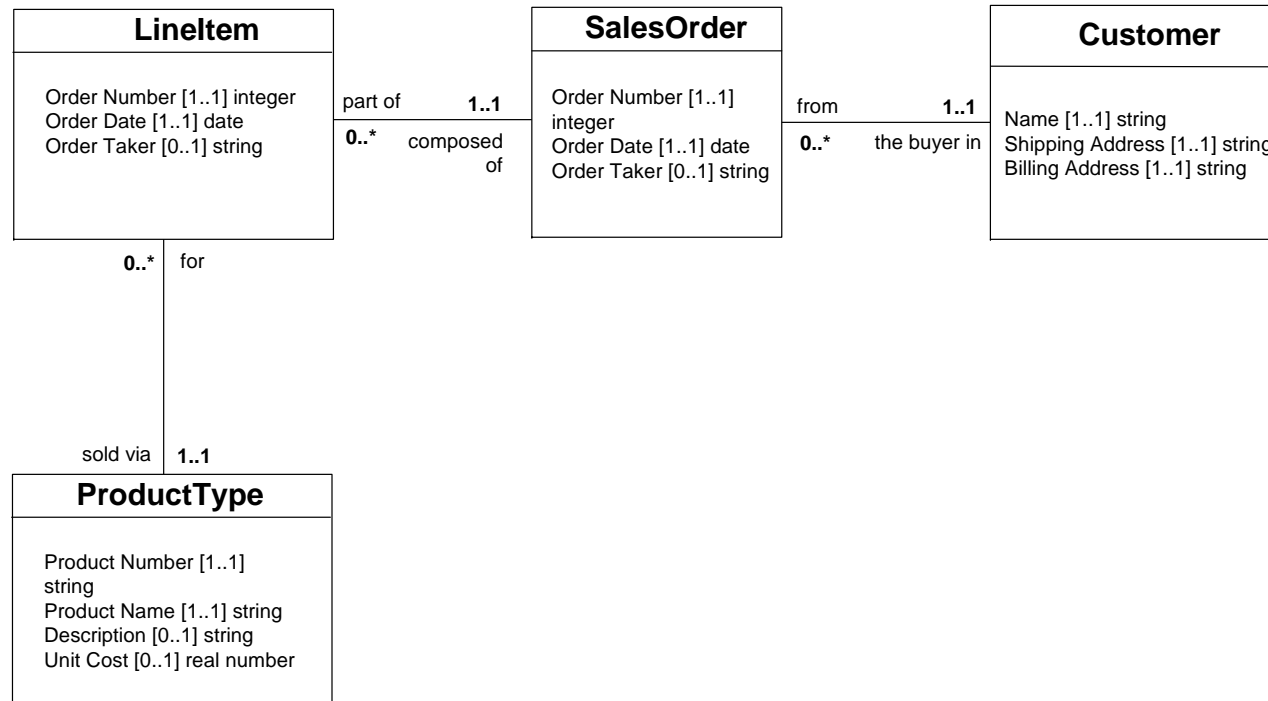


## The Columns . . .

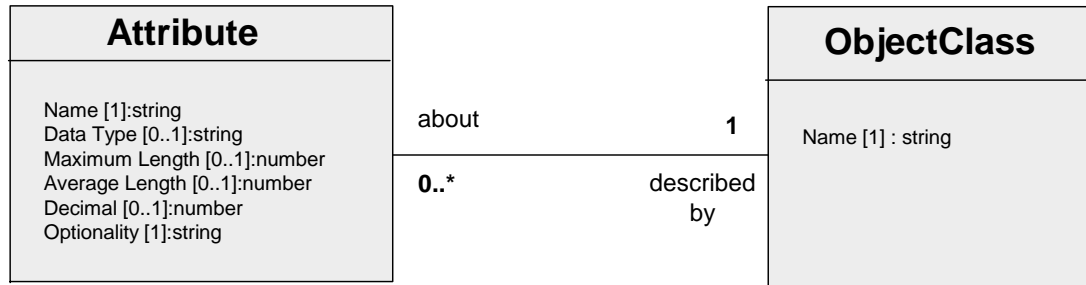
- **Data:** *What* does the enterprise works with?
- **Activities:** *How* does the enterprise do its job?
- **Locations:** *Where* does the enterprise work?
- **People and organizations:** *Who* is involved?
- **Events and timing:** *When* do things happen?
- **Motivation:** *Why* does the enterprise do what it does?



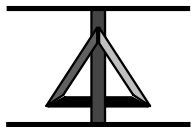
# A typical data (object) model . . .



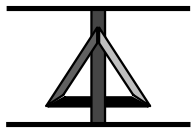
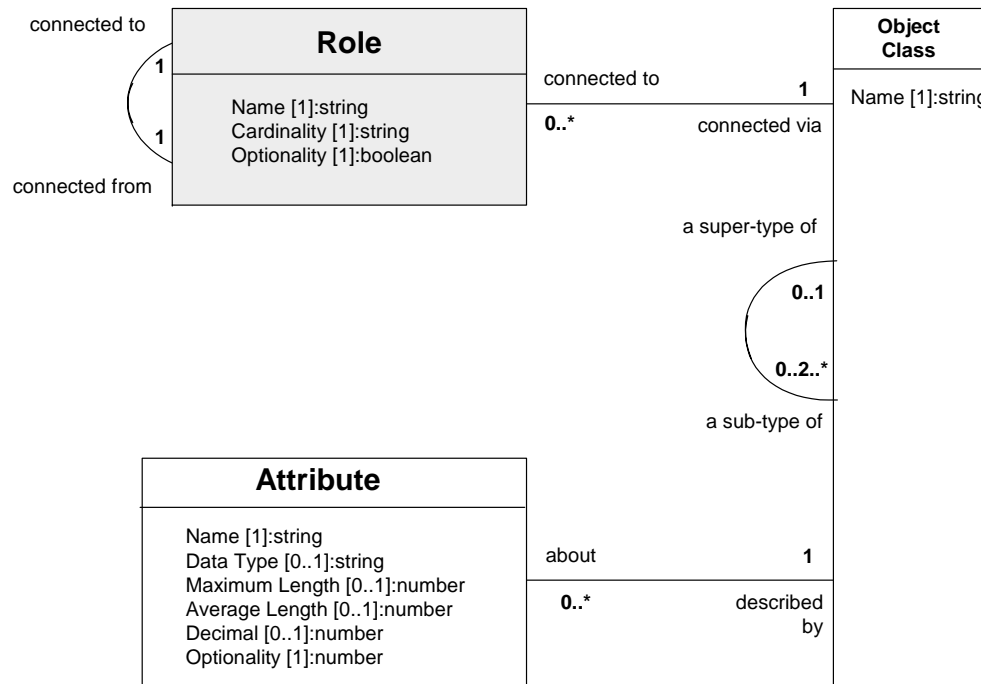
# Our first metamodel . . .



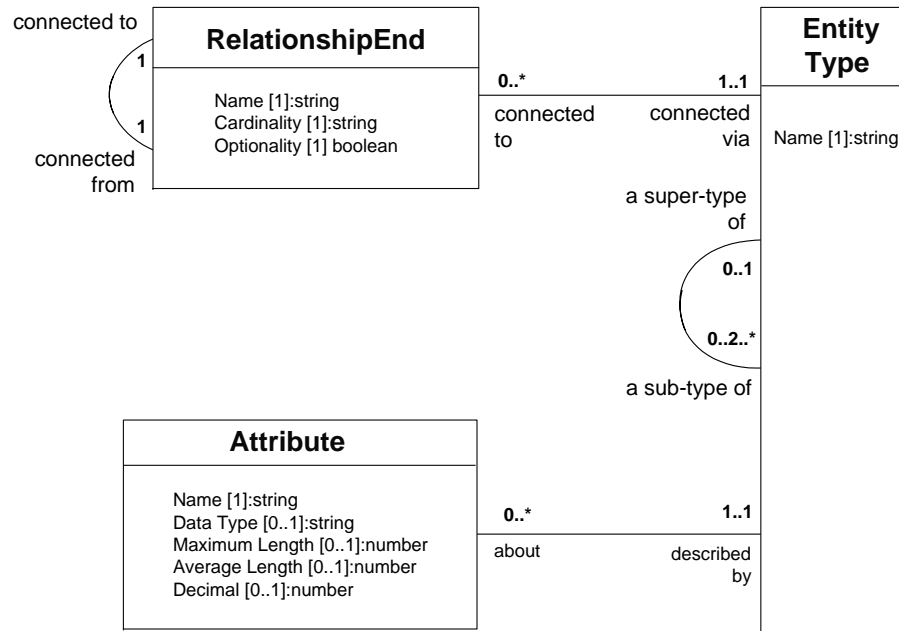
Object Classes	Attributes				
	Name	Object Class (Name)	Name	Data Type	Max. Length
<i>ObjectClass</i>		<i>ObjectClass</i>	<i>Name</i>	<i>String</i>	<i>15</i>
<i>Attribute</i>		<i>Attribute</i>	<i>Name</i>	<i>String</i>	<i>15</i>
		<i>Attribute</i>	<i>Data Type</i>	<i>String</i>	<i>10</i>
		<i>Attribute</i>	<i>Maximum Length</i>	<i>Number</i>	<i>3</i>
		<i>Attribute</i>	<i>...</i>		



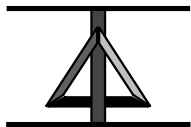
# Add Roles . . .



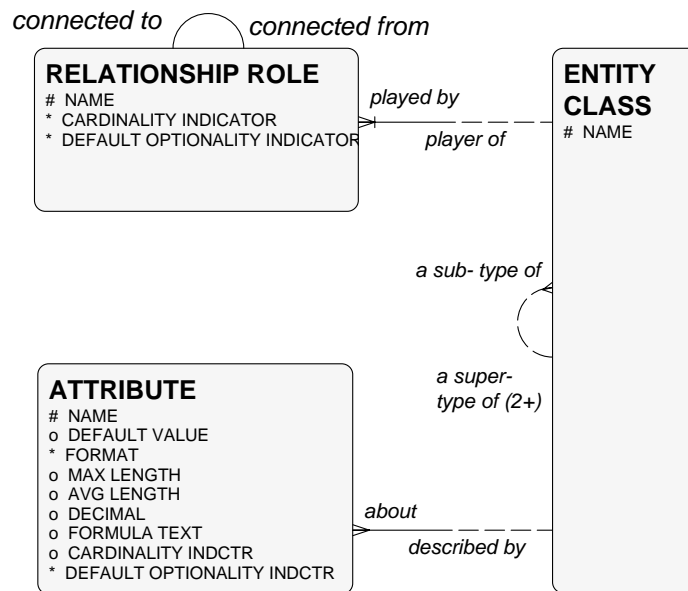
# What about an entity / relationship model, instead . . .



*Strange...It looks just the same...*



# Another approach . . .



Each

<entity class 1>

Must be

(or)

May be

(<relationship name>

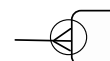
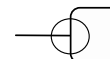
One and only one

(or)

One or more

<entity class 2>

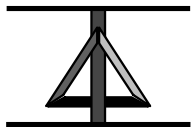
.



For example:

“Each ENTITY CLASS may be *player of* one or more RELATIONSHIP ROLES.”

“Each RELATIONSHIP ROLE must be *played by* one and only one ENTITY CLASS.”



On to the METADATA MODEL . . .

