02291 System Integration

Welcome and Introduction to System Integration

© Giovanni Meroni and Hugo-Andrés López-Acosta

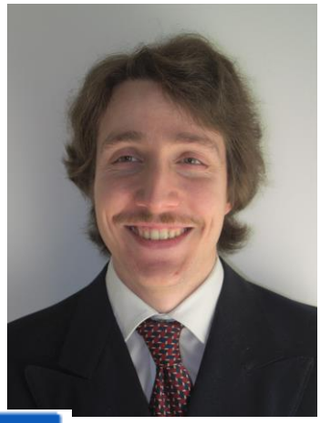
Teachers

- Associate Professor **Hugo-Andrés López-Acosta**, DTU Compute (hulo@dtu.dk)
  - MSc. Software Development Technologies, ITU
  - Ph.D. Computer Science, ITU
  - Postdocs: ITU, DTU, U. Lisbon
  - Industry Experience: Software Consultant (Configit), Product Owner (DCR)
  - Research Interests: BPM, Formal Methods, Distributed Systems, Process Science

https://www.linkedin.com/in/hugoandreslopez/

http://hugo.lopezacosta.net/

- Assistant Professor **Giovanni Meroni**, DTU Compute (giom@dtu.dk)
  - MSc in Computer Engineering, Politecnico di Milano 2013
  - PhD in Information Engineering, Politecnico di Milano 2018
  - IT Consultant at Reti S.p.A. 2013 – 2014
  - Postdoctoral Research Assistant at Politecnico di Milano 2018 - 2022
  - Researcher & teacher at DTU since 2022
  - Research Interests: BPM, Information Systems, Process Monitoring, SOA



https://www.linkedin.com/in/giovanni-meroni-18b1922a9/

https://orbit.dtu.dk/en/persons/giovanni-meroni

Learning Objectives

- From https://kurser.dtu.dk/course/02291
- A student who has met the objectives of the course will be able to:
  - analyze a problem and model the requirements of a software system
  - model the design of a software system
  - formulate safety and liveness properties of models
  - analyze and validate structural and semantic properties in a model
  - apply modeling techniques for domain modeling, requirements, design, and systems in a specific case
  - model in a team
  - discuss the appropriateness of models to explain a given problem
  - explain a socio-technical system using model-driven frameworks

Course Material

- A selection of book chapters and articles freely available to DTU students
  – To access them, you need to access the campus network (in presence or through VPN)
- For each course topic, associated material will be listed at the end of the lecture’s slides
- A complete list will be available on DTU Learn

Course Activities

- Activities
  - Lectures: most (but not all) Wednesdays 8:00-9:45
    - Lectures will be streamed via MS Teams and recorded.
    - Slides and video recordings can be found on DTU Learn Content.
  - Exercise sessions: Wednesdays 10:00-11:45
    - Mainly devoted to solve exercises and work on group project
  - Mandatory group project
    - Covers most course topics
    - Groups of 6 students
  - Written exam: May 29, 2024
- Assessment:
  - An approved group project is mandatory for participation in the written examination!
  - The final mark is the mark from the written examination.

(tentative) Lecture Plan

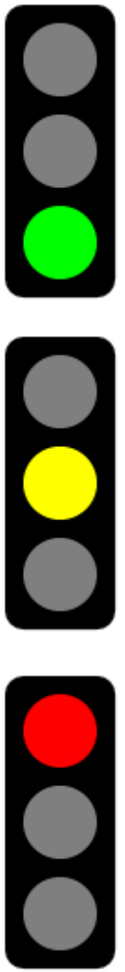
Date	Topic	Teacher
31/01	Welcome	HL and GM
31/01	Goal-oriented Requirements Engineering	GM
07/02	Recap on Formal Methods	HL
14/02	ArchiMate	GM
21/02	Behavioural Models	GM
28/02	Distributed Systems Models	HL
06/03	Mandatory Group Project	N/A
13/03	Timed Automata	HL
20/03	Introduction to BPMN	GM
Easter break		
03/04	Advanced BPMN	GM
10/04	DCR Graphs (part 1)	HL
17/04	DCR Graphs (part 2)	HL
24/04	Decision Models	GM
01/05	Wrap-up	HL and GM

About Group Registration

- We have 300+ students, so you must work together and help each other in groups.
- You should form groups of 6 persons for the exercise sessions the rest of the semester and the mandatory group project.
- Look for group partners:
  - contact people you know, or
  - talk with people you meet at DTU in the exercise session today, or
  - use the Discussion Forum "Look for Group Partners“ on DTU Learn
- When you have found a group, you should register it on DTU Learn. Must be done by end of Wednesday 14th of February.

Group project outcome

- Pass:
  - Your project may have a few non-fatal flaws, but overall is ok
  - You are admitted to the final exam
  - Your preparation *should* be fine for passing the final exam
- Pass with warnings:
  - Your project is borderline
  - You are admitted to the final exam
  - You *need* to improve your preparation.
    - Otherwise, chances of passing the final exam are low
- Fail:
  - Your project has several fatal flaws
  - You are not admitted to the final exam
  - You will have to re-take the course next year



Support and feedback

- During lectures and exercise sessions, you can ask questions to the lecturer
  - If you cannot attend in person, you can use MS Teams
  - Priority will be given to questions asked in person
- Outside class hours, you can ask and answer questions on the Q&A Forum on DTU Learn
  - You are highly invited to contribute to the forum (you can post anonymously if you wish)
  - The question you have may already have been answered there
  - In the past we had very nice interactions there
  - Questions asked via MS Teams outside class hours will be ignored
- Email is discouraged as a communication channel, since it does not scale with 300+ students

Motivation

Why is Software Complex?

Let us say software is complex if it is hard to understand and modify

Accidental (or Incidental) complexity is tied to an implementation

It can be removed by cleaning up the code, changing the language, etc.

Intrinsic complexity remains even if all accidental complexity is removed

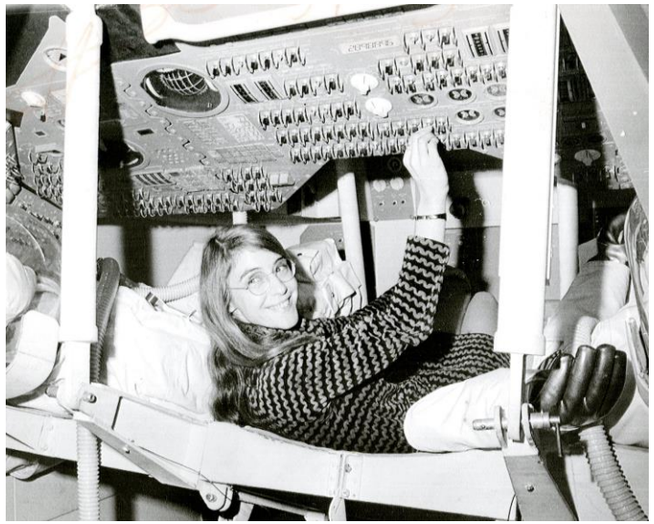
It can be managed by designing run-time behavior more predictable

Software Engineering

- Is the **systematic** application of engineering approaches to the development and building of software

To reduce incidental and accidental complexity

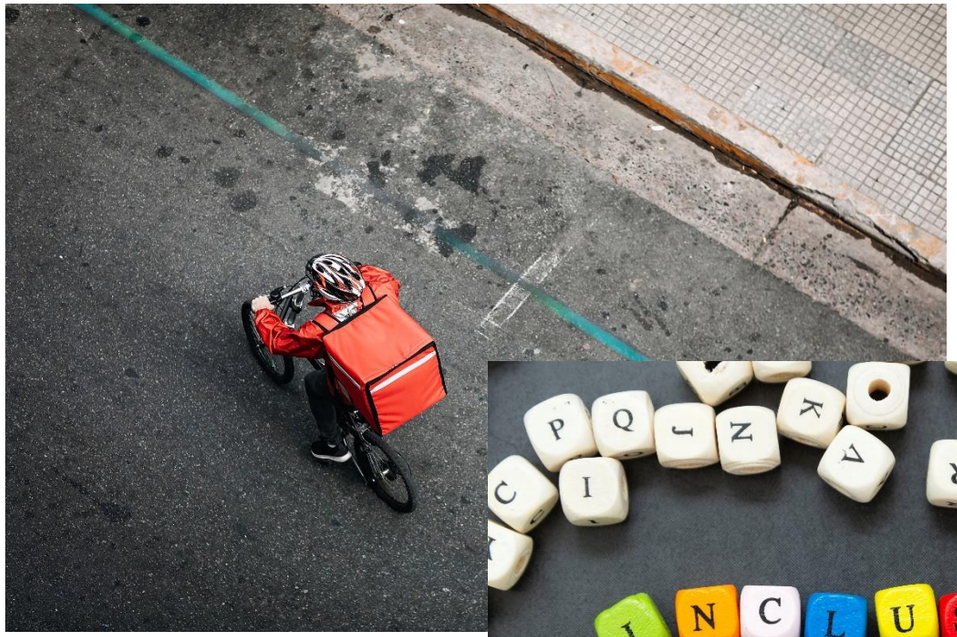
- We will focus on **Model-Driven Software Engineering**, that is, the elicitation, analysis, and execution of software via models

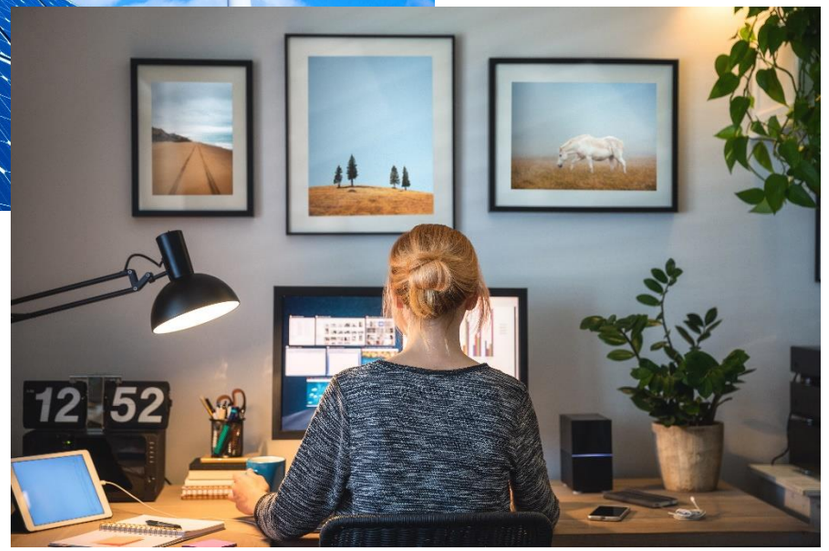


Margaret Hamilton, former director of the Software Engineering Division of the MIT Instrumentation Laboratory, which developed on-board flight software for NASA's Apollo program.

Real-world applications are complex

- Need to cope with ever-changing business goals



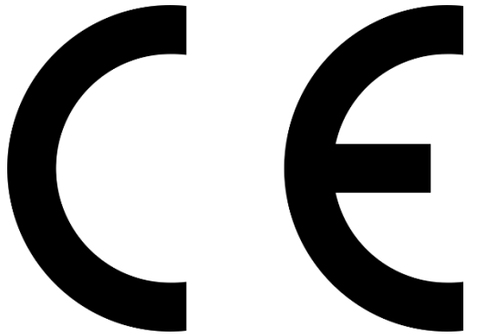


Real-world applications are complex

- Need to interact with different types of users

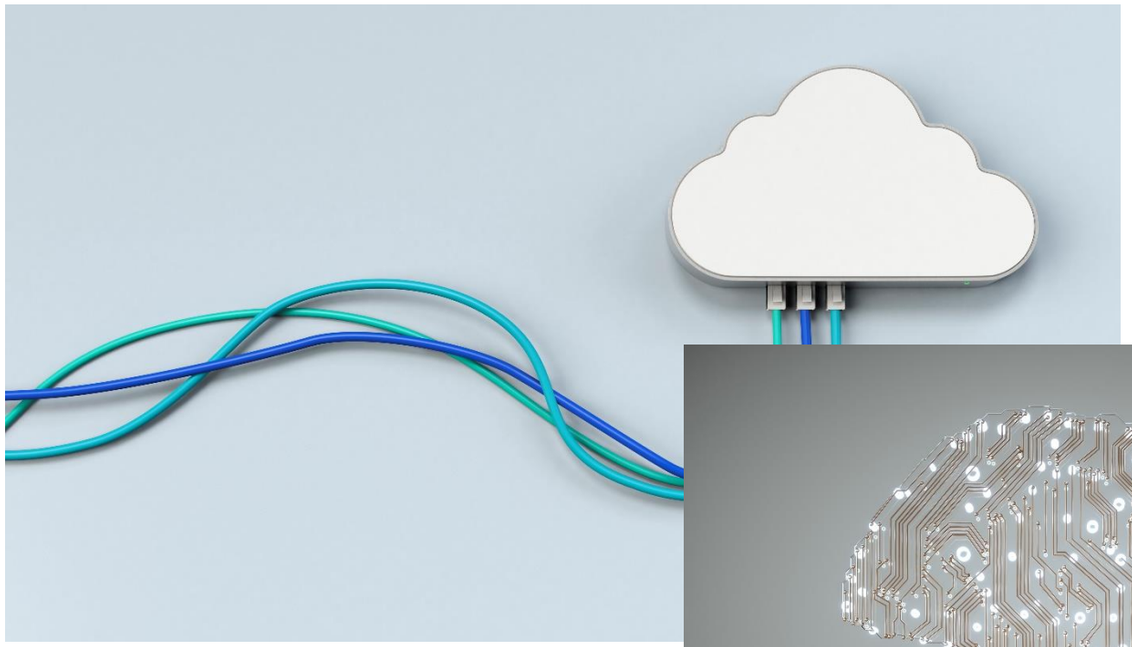
Real-world applications are complex

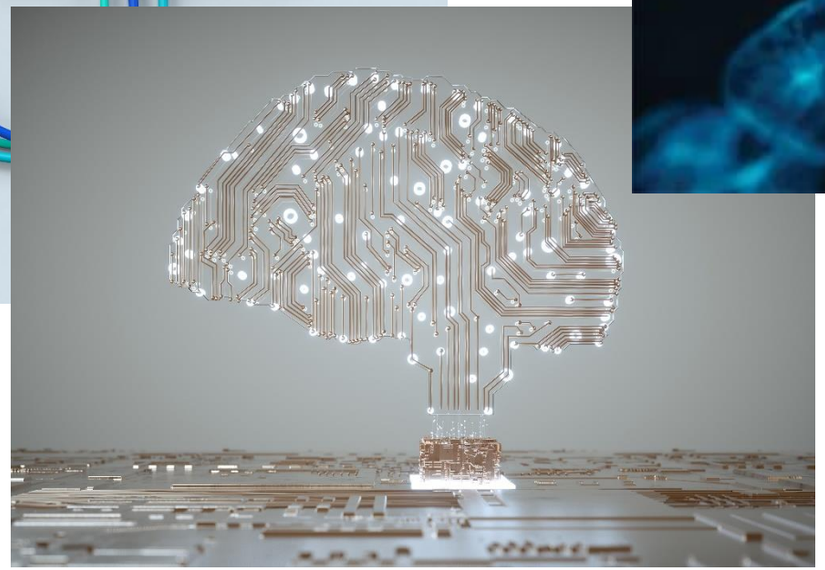
- Need to comply with security, safety and privacy regulations



Real-world applications are complex

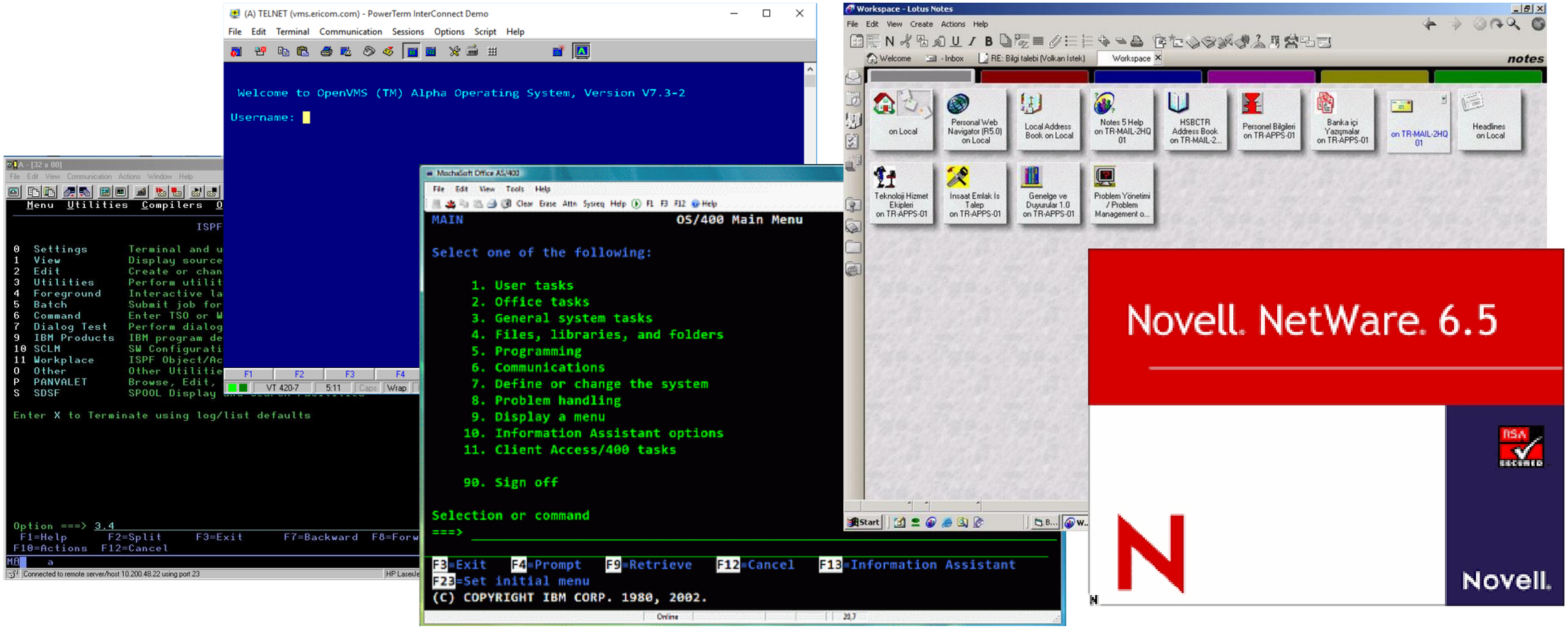
- Need to consider emerging technology





Real-world applications are complex

- Need to integrate with legacy applications and systems



Software Engineering and Complexity

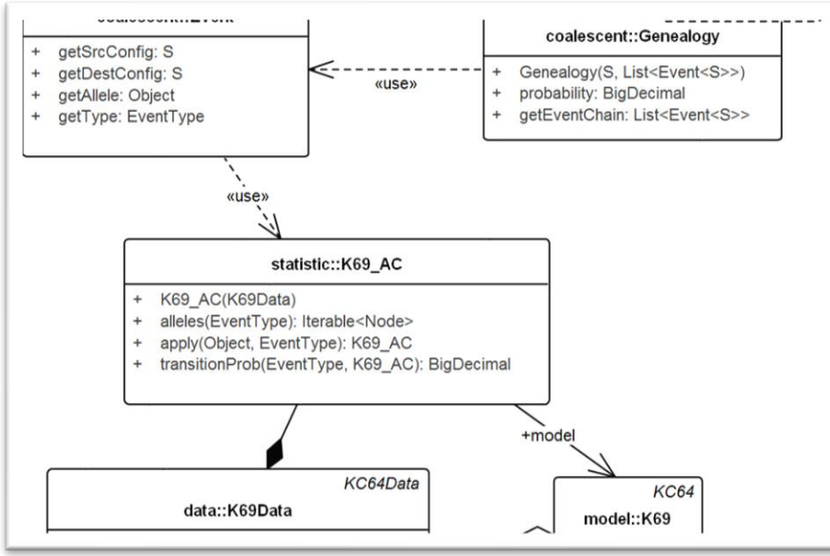
- Avoid Accidental (and Incidental) complexity
  - Use the right languages, frameworks, and tools for the job
  - Avoid technical debt, which accumulates, and makes code brittle
  - Technical debt is the conceptual cost of cleaning up a “quick fix”
- Manage Intrinsic complexity
  - Design modular systems, with relatively independent parts

Modelling for Software Development

- We rely on software models due to four facts:
  - Software artifacts are becoming more complex and they need to be discussed at different abstraction levels, and depending on the stakeholders, software development phases and objectives
  - Software is pervasive on people's lifes, and the need of new pieces, or changes in the existing behaviour is a constant
  - Shortage of SW development skills wrt demand
  - Software Development is far from "just coding", it requires interaction with no-experts, other professions, or members of the team that have never seen your code

The use of models in Software Engineering[1]

- **Models as sketches**: for communication purposes, only partial views of the system are given
  - E.g.: UML class and sequence diagrams
- **Models as blueprints**: used to provide a complete and detailed specification of the system
  - E.g.: BPMN models “mined” from an ERP system
- **Models as programs**: instead of code, they are used to develop the system
  - E.g.: BPMN models “plugged-in” an execution engine (for example, Camunda)

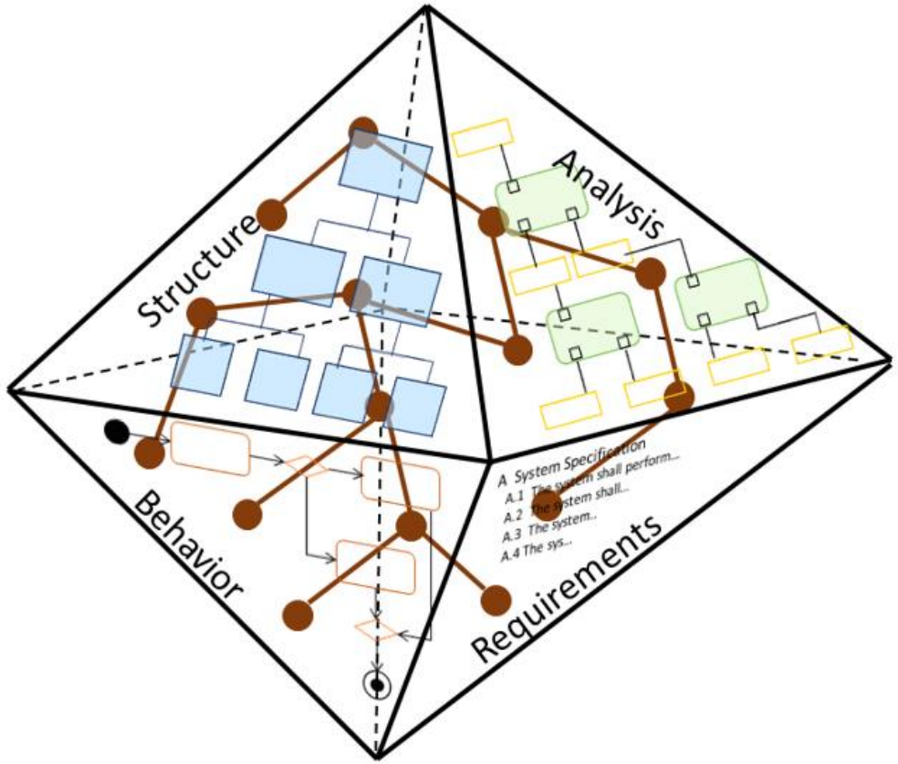


[1] https://martinfowler.com/bliki/UmlMode.html

System Integration

- Def. 1.: “the process of bringing together the component subsystems into one system (an aggregation of subsystems cooperating so that the system is able to deliver the overarching functionality) and ensuring that the subsystems function together as a system”

- Def. 2. “as the process of linking together different computing systems and software applications physically or functionally to act as a coordinated whole”



© Dirk Zwemer. MBSE and Integration.
https://intercax.com/2020/02/20/mbse-and-integration-part-1/

Enterprise Architectures

Enterprise architecture

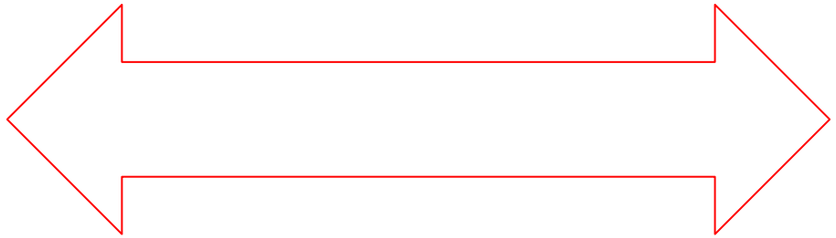
Enterprise architecture (EA) is "a well-defined practice for conducting enterprise analysis, design, planning, and implementation, using a holistic approach at all times, for the successful development and execution of strategy. Enterprise architecture applies architecture principles and practices to guide organizations through the business, information, process, and technology changes necessary to execute their strategies. These practices utilize the various aspects of an enterprise to identify, motivate, and achieve these changes"

Federation of EA Professional Organizations, Common Perspectives on Enterprise Architecture, Architecture and Governance Magazine, Issue 9-4, November 2013 (2013). Retrieved on November 19, 2013

EA bridges Strategy and Implementation

Business strategy

Business drivers
Business goals
Business policy
Trend Analysis



Architecture

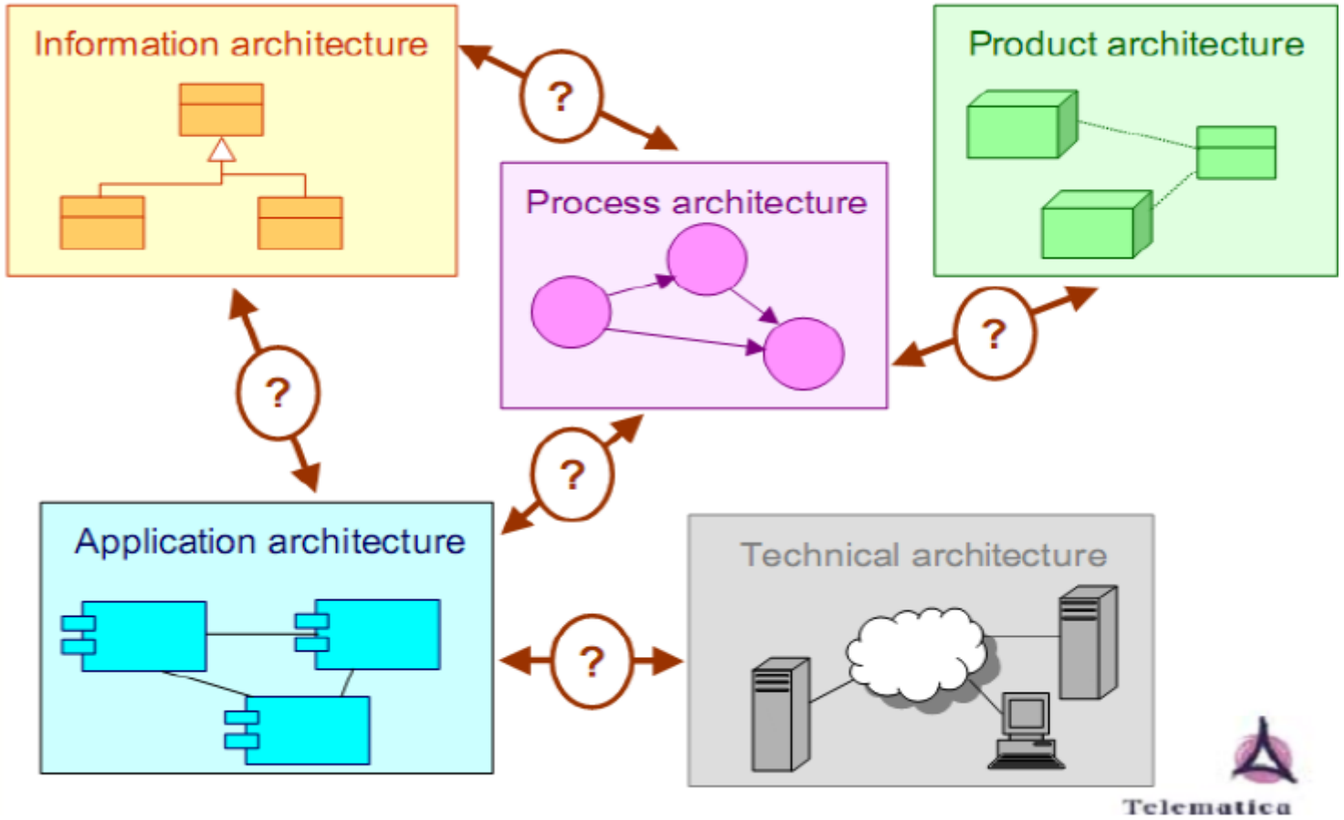
Business architecture
Information architecture
Solution architecture
Technology architecture

Implementation

Business processes
Application systems
Tech infrastructure
Organizational structure

Sobah Abbas Petersen “Introduction to Enterprise Architecture” 2012

Enterprise Architecture: describing coherence



The enterprise view

- Why do this at the enterprise level?
  - Look at “the whole,” not the parts
  - Look beyond narrow and restricted views
  - Look for context from the top

- The quality of all IT decisions is dependent on the enterprise view

Zachman Framework

- Zachman framework was proposed in 1987
- It provides a taxonomy for organizing architectural artifacts, that are documents, specifications, and models
- Such taxonomy considers artifact targets (specific users) and the aspect (e.g., data or network) is being addressed

Orand and Villarreal “Foundations of IT service Management with ITIL 2011”

Zachman Framework

	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organization Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	List of Business Goals/Stratgies Ends/Means = Major Business Goal/Strategy	SCOPE (CONTEXTUAL) *Planner*
BUSINESS MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	BUSINESS MODEL (CONCEPTUAL) *Owner*
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. Application Architecture Proc. = Application Function I/O = User Views	e.g. Distributed System Architecture Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means = Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. System Design Proc.= Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture Node = Hardware/Systems Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. Program Proc.= Language Statement I/O = Control Block	e.g. Network Architecture Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

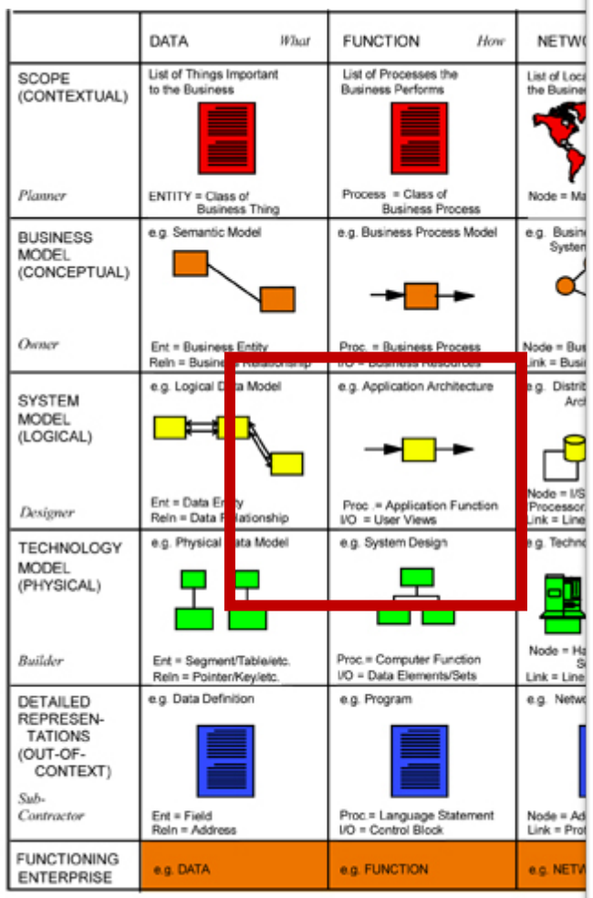
Zachman Framework

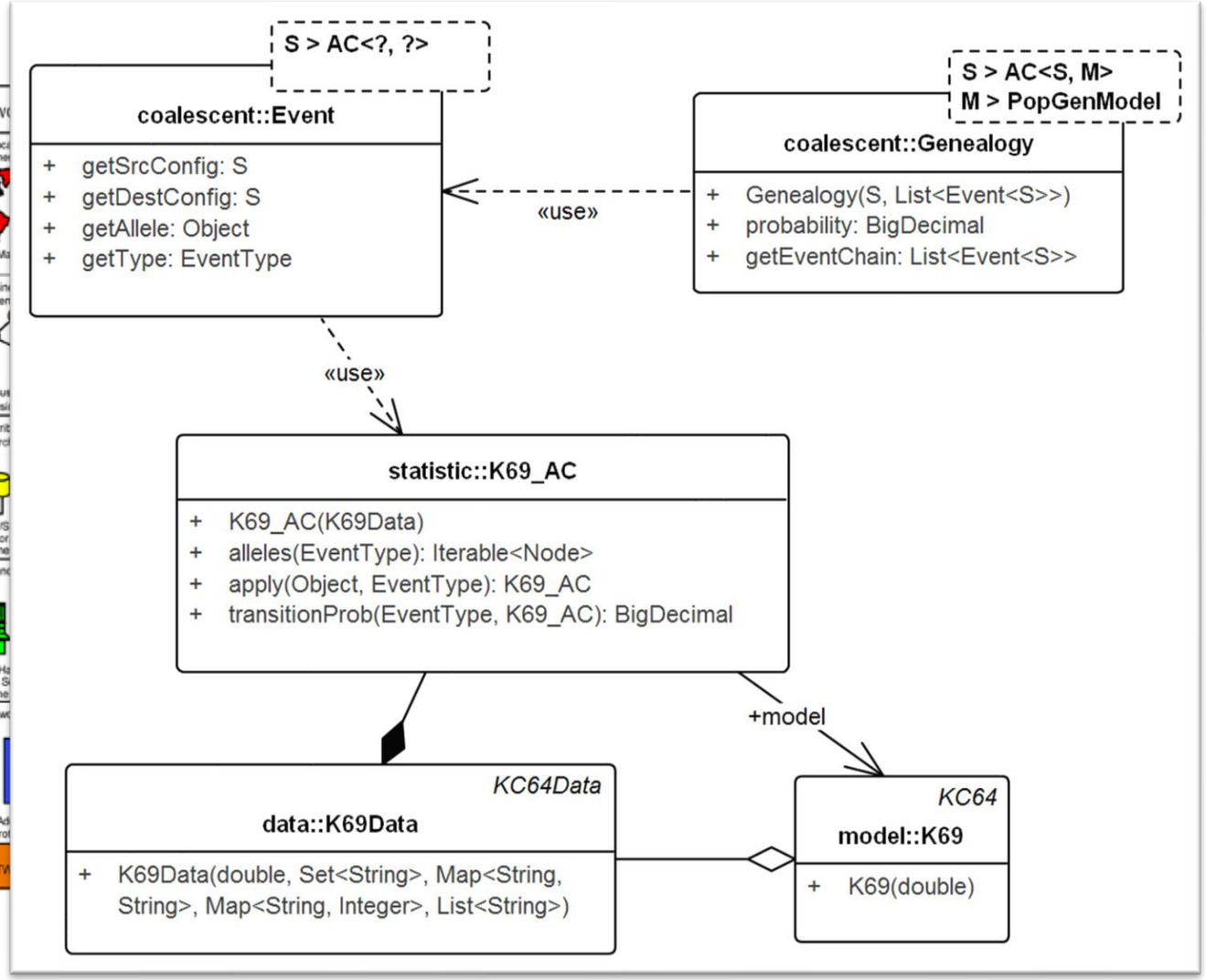
- Artifacts are organized along 36 categories
- Categories can be grouped in 6 perspectives, depending on the level of detail:
  - Scope: high-level objectives, size, shape, and relationships of the enterprise
  - Enterprise Model: conceptual business model, processes and business goals
  - System Model: technology-independent software components, data models, workflows, web services
  - Technology Model: software and network topology, technology-dependent software components, data models, workflows, web services
  - Detailed Representation: source code, databases and executable models
  - Operational model: actual software, people and data

Zachman Framework

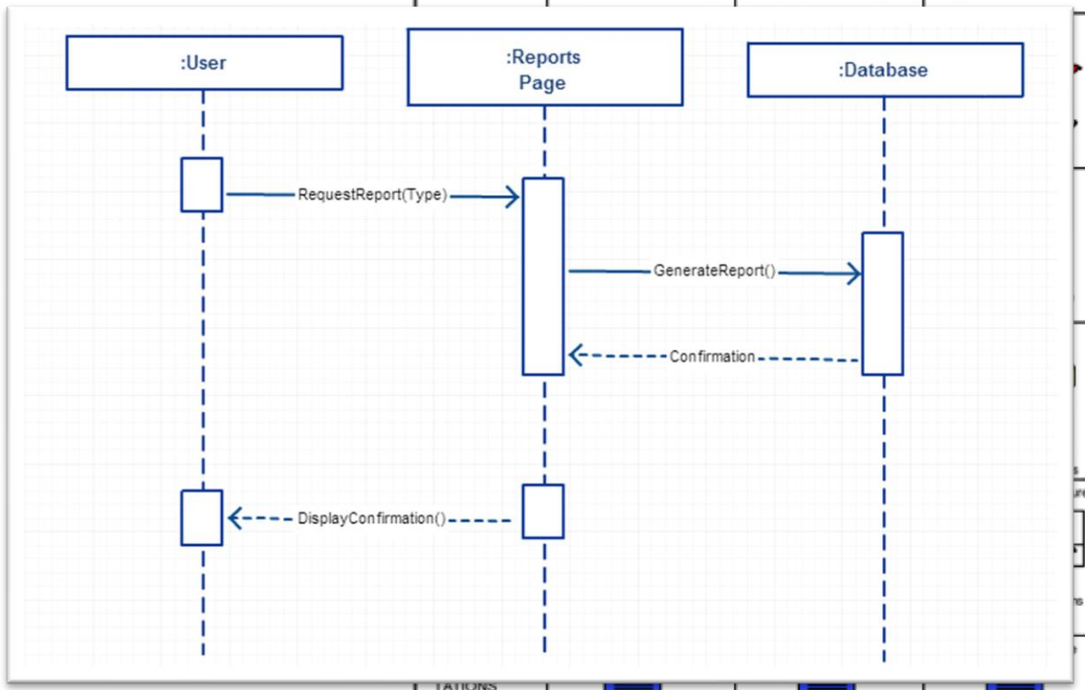
- Categories can be grouped in 6 viewpoints, based on the questions they help at answering:
  - Data (what is processed?)
  - Function (how is processed?)
  - Network (where is processed?)
  - People (who needs it?)
  - Time (when is processed?)
  - Motivation (why is processed?)
- You may have already seen artifacts addressing some of the categories in Zachman Framework

UML Class Diagrams





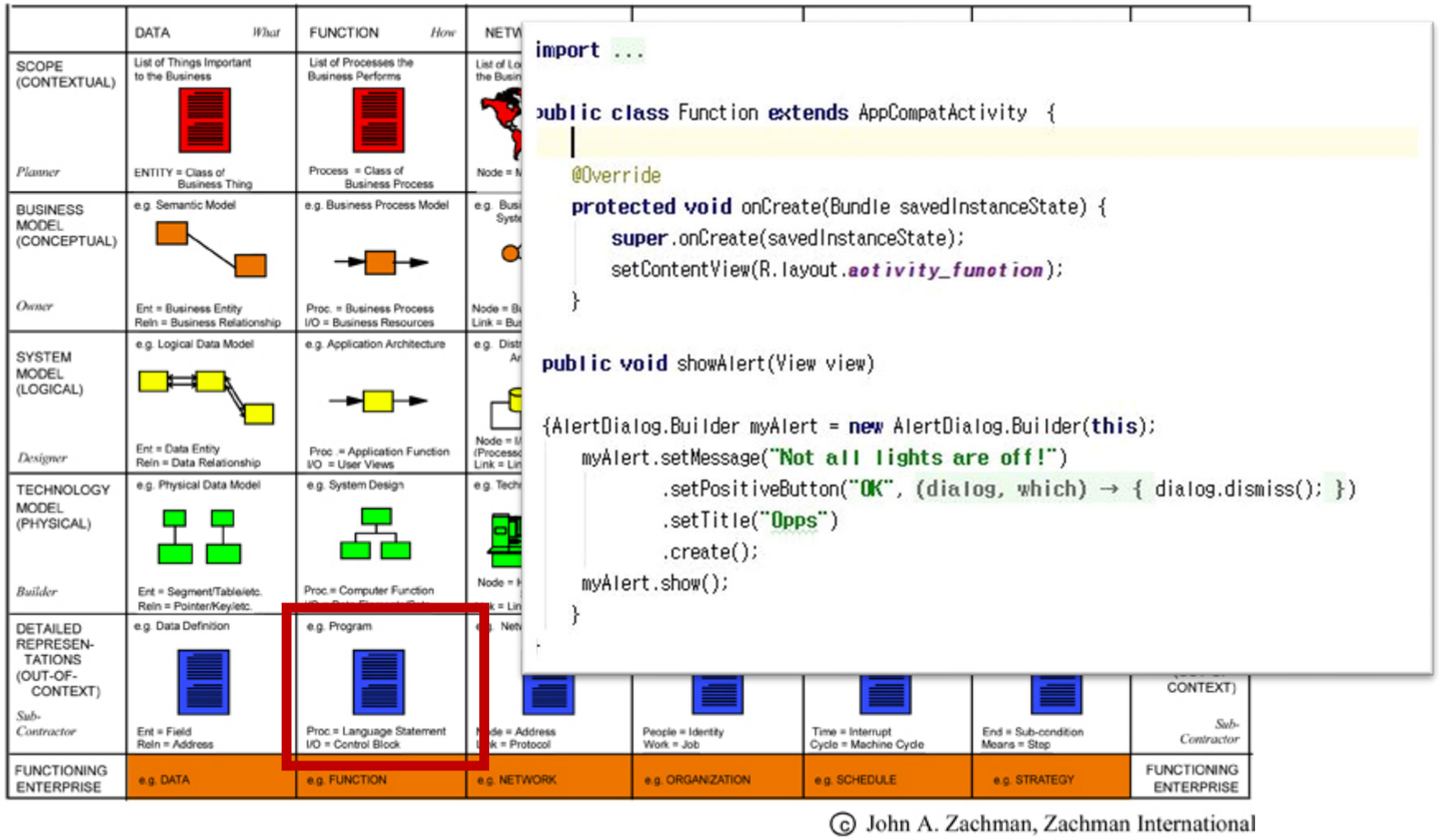
UML Sequence Diagrams



	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
				List of Organizations Important to the Business	List of Events/Cycles Significant to the Business	List of Business Goals/Strategies	SCOPE (CONTEXTUAL)
				People = Major Organization Unit	Time = Major Business Event/Cycle	Ends/Means = Major Business Goal/Strategy	*Planner*
				e.g. Work Flow Model	e.g. Master Schedule	e.g. Business Plan	BUSINESS MODEL (CONCEPTUAL)
				People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	End = Business Objective Means = Business Strategy	*Owner*
				e.g. Human Interface Architecture	e.g. Processing Structure	e.g. Business Rule Model	SYSTEM MODEL (LOGICAL)
				People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	End = Structural Assertion Means =Action Assertion	*Designer*
				e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design	TECHNOLOGY MODEL (PHYSICAL)
				People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	End = Condition Means = Action	*Builder*
TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	Ent = Field Reln = Address	Proc = Language Statement I/O = Control Block	Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

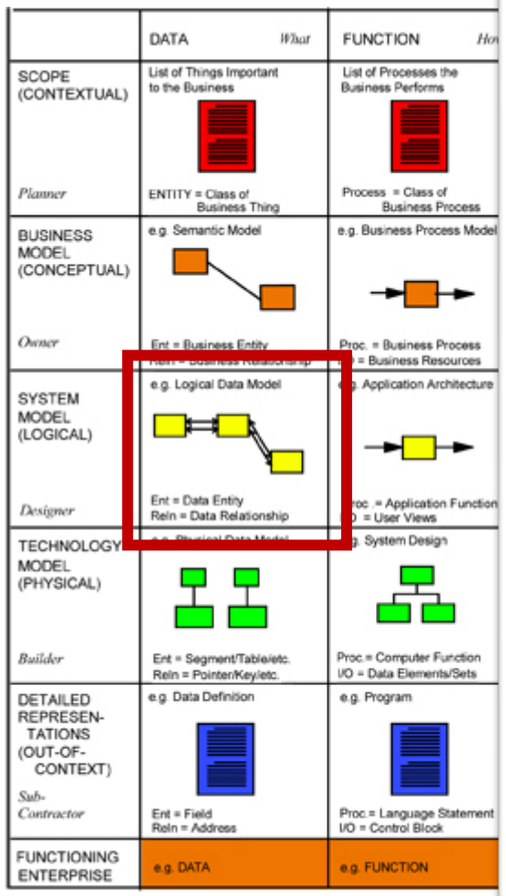
© John A. Zachman, Zachman International

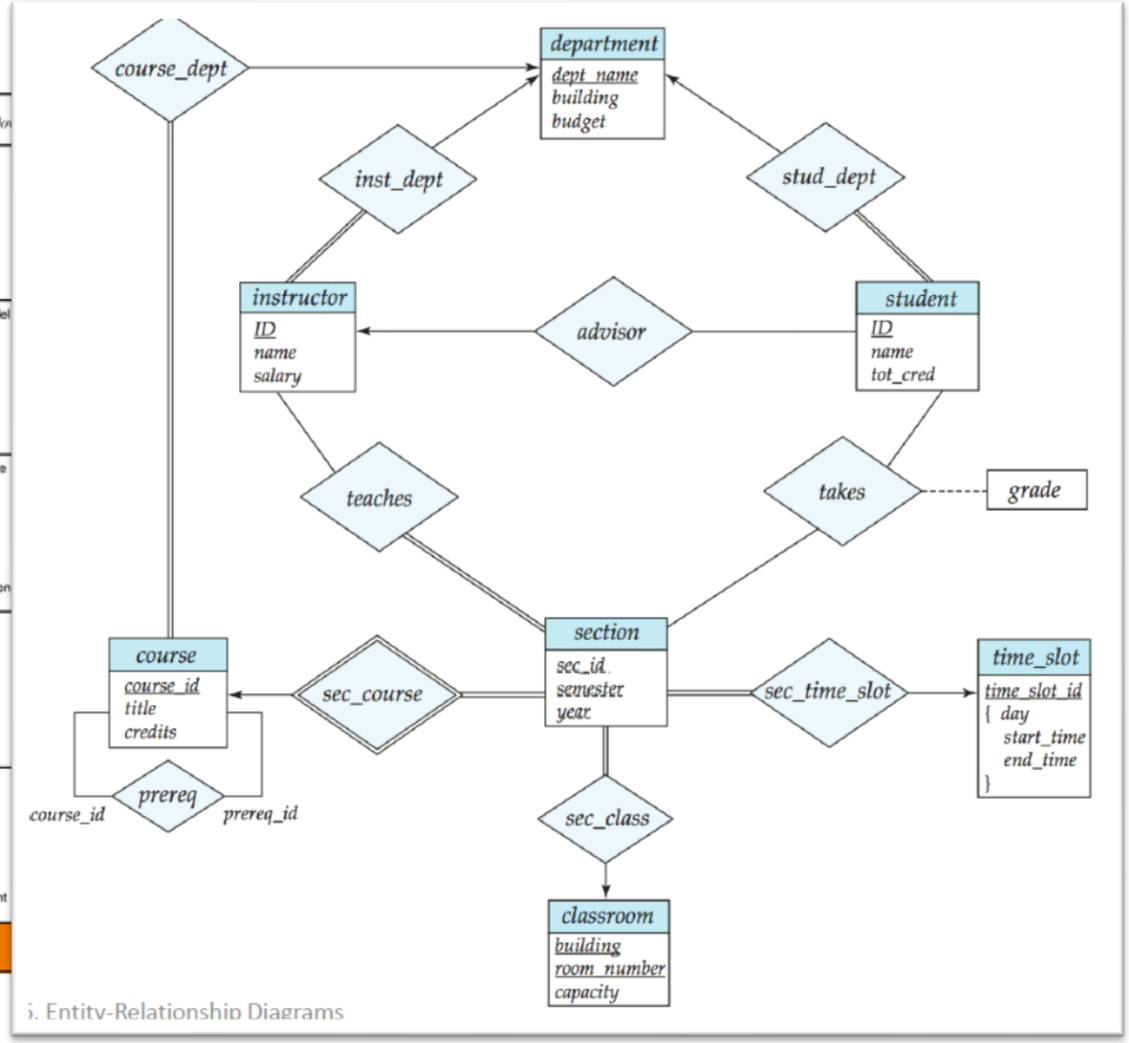
Java source code



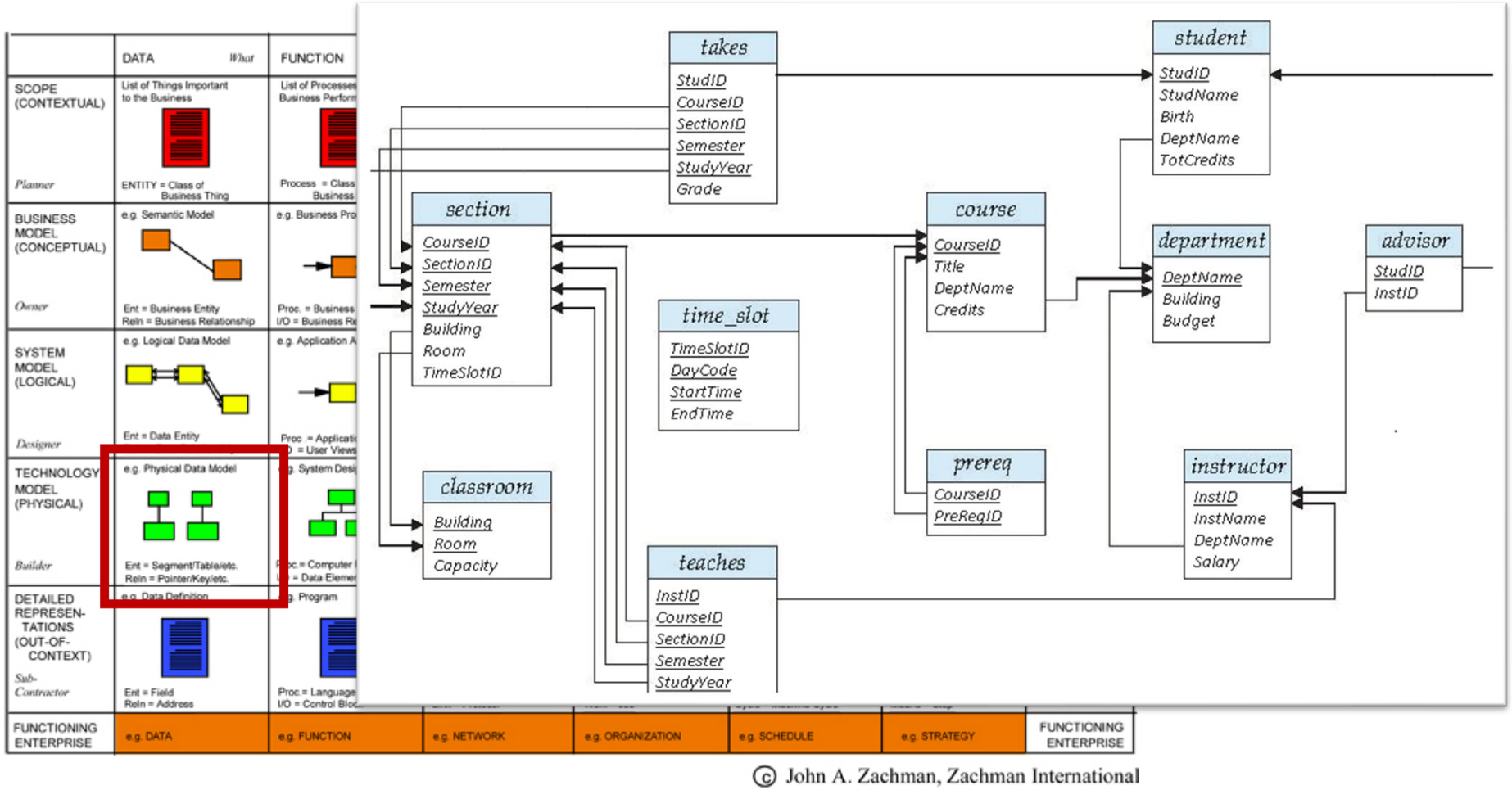
© John A. Zachman, Zachman International

ER Diagrams


DATA
What
FUNCTION
SCOPE (CONTEXTUAL)
List of Things Important to the Business
List of Processes the Business Performs
Planner
ENTITY = Class of Business Thing
Process = Class of Business Process
BUSINESS MODEL (CONCEPTUAL)
e.g. Semantic Model
e.g. Business Process Model
Owner
Ent = Business Entity
Proc. = Business Process
SYSTEM MODEL (LOGICAL)
e.g. Logical Data Model
Designer
Ent = Data Entity
Reln = Data Relationship
TECHNOLOGY MODEL (PHYSICAL)
e.g. System Design
Builder
Ent = Segment/Table/etc.
Reln = Pointer/Key/etc.
Proc.= Computer Function
I/O = Data Elements/Sets
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT)
e.g. Data Definition
e.g. Program
Sub-Contractor
Ent = Field
Reln = Address
Proc.= Language Statement
I/O = Control Block
FUNCTIONING ENTERPRISE
e.g. DATA
e.g. FUNCTION


course_dept
department
dept_name
building
budget
inst_dept
stud_dept
instructor
ID
name
salary
advisor
student
ID
name
tot_cred
teaches
takes
grade
course
course_id
title
credits
sec_course
section
sec_id
semester
year
sec_time_slot
time_slot
time_slot_id
{ day
start_time
end_time
}
prereq
course_id
prereq_id
sec_class
classroom
building
room_number
capacity
). Entity-Relationship Diagrams

Database Schema Diagrams



SQL Data Definition Language

```
2
3  CREATE TABLE `navigatie` (
4    `navigatie_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
5    `huidige_vraaggroep` varchar(255) NOT NULL,
6    `vorige_vraaggroep` varchar(255) DEFAULT NULL,
7    `richting` varchar(255) NOT NULL,
8    `datum_en_tijd` timestamp(3) NOT NULL DEFAULT CURRENT_TIMESTAMP(3) ON UPDATE CURRENT_TIMESTAMP(3),
9    `schadegeval_id` bigint(20) unsigned DEFAULT NULL,
10   `claim_id` bigint(20) unsigned DEFAULT NULL,
11   `gebruiker_id` bigint(20) NOT NULL,
12   `soort_gebruiker` varchar(255) NOT NULL,
13   PRIMARY KEY (`navigatie_id`)
14 ) ENGINE=InnoDB DEFAULT CHARSET=utf8$$
15
16
```

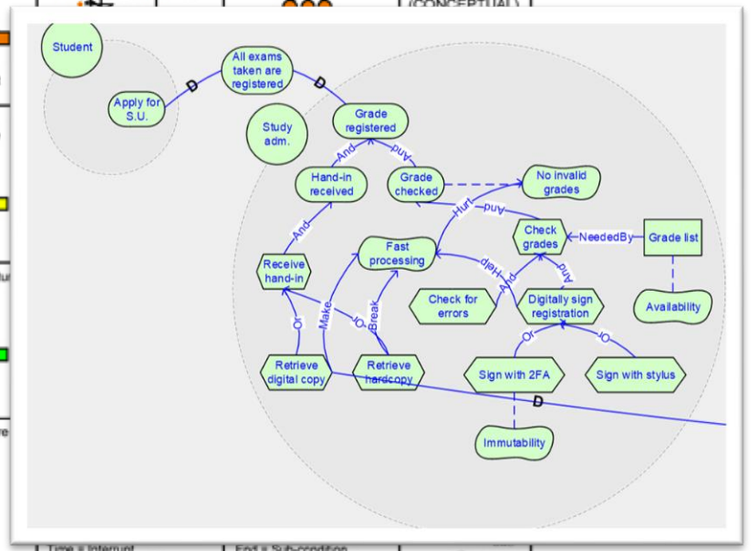
	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL)	List of Things Important to the Business	List of Processes the Business Performs	List of Locations in which the Business Operates	List of Organizations Important to the Business	List of Events/Cycles Significant to the Business	List of Business Goals/Strategies	SCOPE (CONTEXTUAL)
Builder	Reln = Pointer/Key/etc.	Proc.= Computer Function = Data Elements/Sets	Node = Hardware/Systems Software Link = Line Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	End = Condition Means = Action	*Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. Program Proc.= Language Statement = Control Block	e.g. Network Architecture Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

© John A. Zachman, Zachman International

What you will learn

I-star (today)

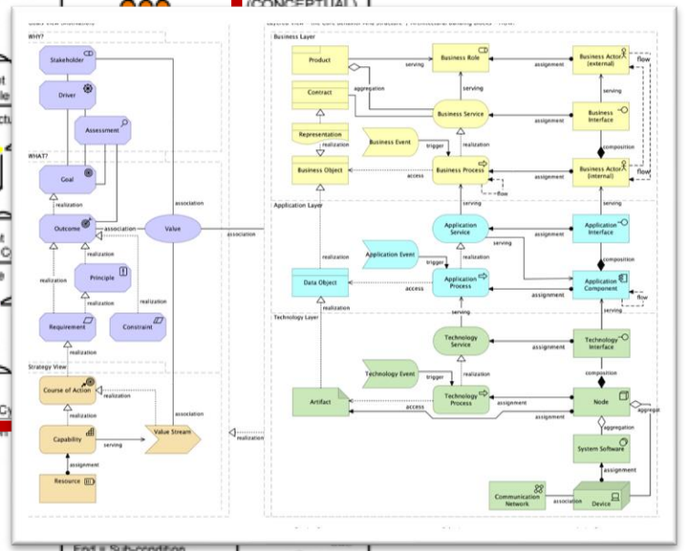
	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organization Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	List of Business Goals/Strategies Ends/Means = Major Business Goal/Strategy	SCOPE (CONTEXTUAL) *Planner*
BUSINESS MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Rein = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule	e.g. Business Plan	BUSINESS MODEL (CONCEPTUAL)
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Rein = Data Relationship	e.g. Application Architecture Proc. = Application Function I/O = User Views	e.g. Distributed System Architecture Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable			
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Rein = Pointer/Key/etc.	e.g. System Design Proc.= Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture Node = Hardware/Systems Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format			
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Rein = Address	e.g. Program Proc.= Language Statement I/O = Control Block	e.g. Network Architecture Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Sub-condition Means = Step	*Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE



© John A. Zachman, Zachman International

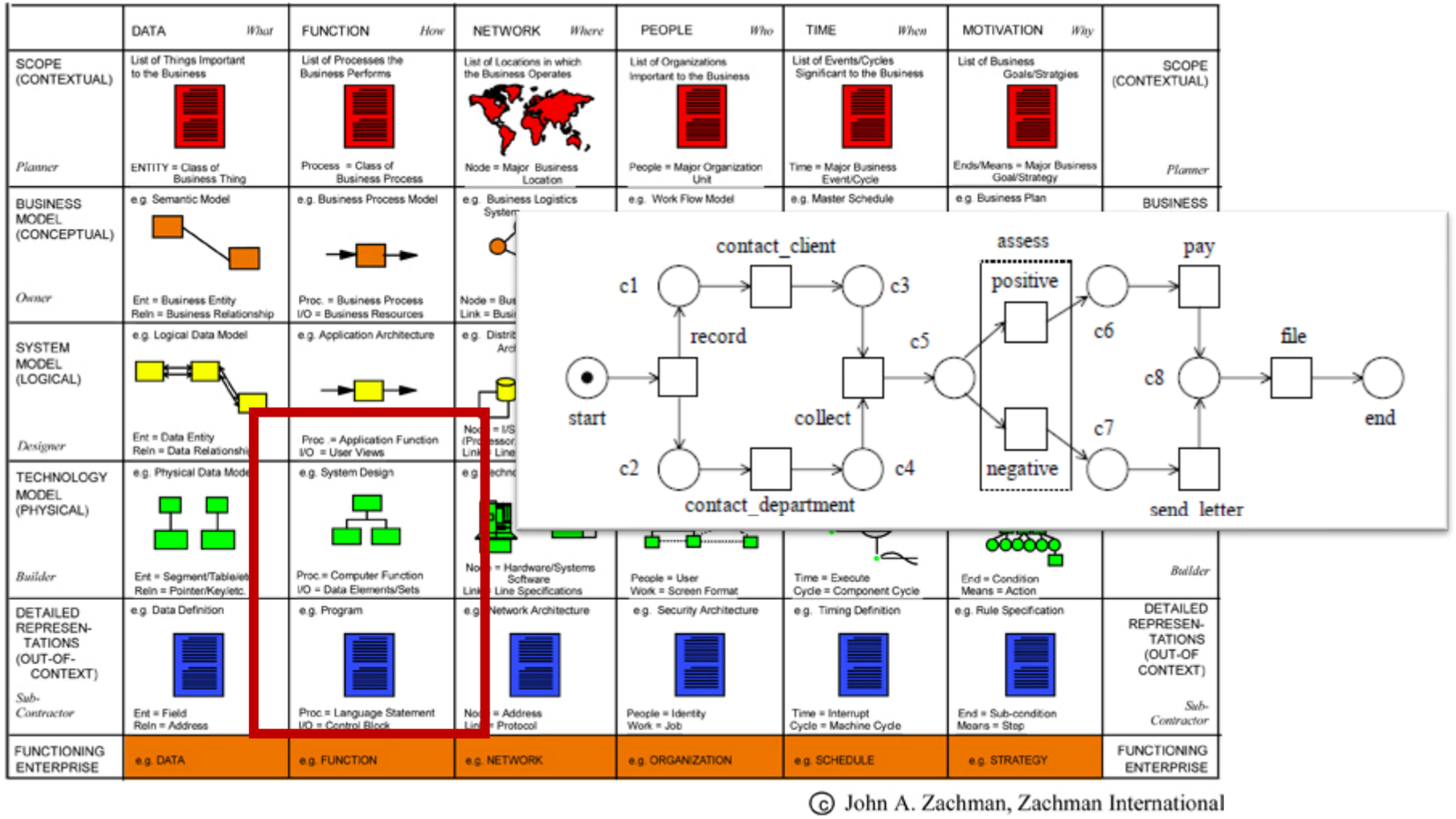
ArchiMate (today and week 7)

	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	to the Business	Business Performs	the Business Operates	Important to the Business	Significant to the Business	Goals/Strategies	SCOPE (CONTEXTUAL) *Planner*
	ENTITY = Class of Business Thing	Process = Class of Business Process	Node = Major Business Location	People = Major Organization Unit	Time = Major Business Event/Cycle	Ends/Means = Major Business Goal/Strategy	
BUSINESS MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Work Flow Model	e.g. Master Schedule	e.g. Business Plan	BUSINESS MODEL (CONCEPTUAL)
	Ent = Business Entity Reln = Business Relationship	Proc. = Business Process I/O = Business Resources	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle		
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Processing Struct		
	Ent = Data Entity Reln = Data Relationship	Proc. = Application Function I/O = User Views	Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing C		
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure		
	Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	Proc. = Computer Function I/O = Data Elements/Sets	Node = Hardware/Systems Software Link = Line Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cy		
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*							*Contractor*
	Ent = Field Reln = Address	Proc. = Language Statement I/O = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Sub-condition Means = Step	
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE



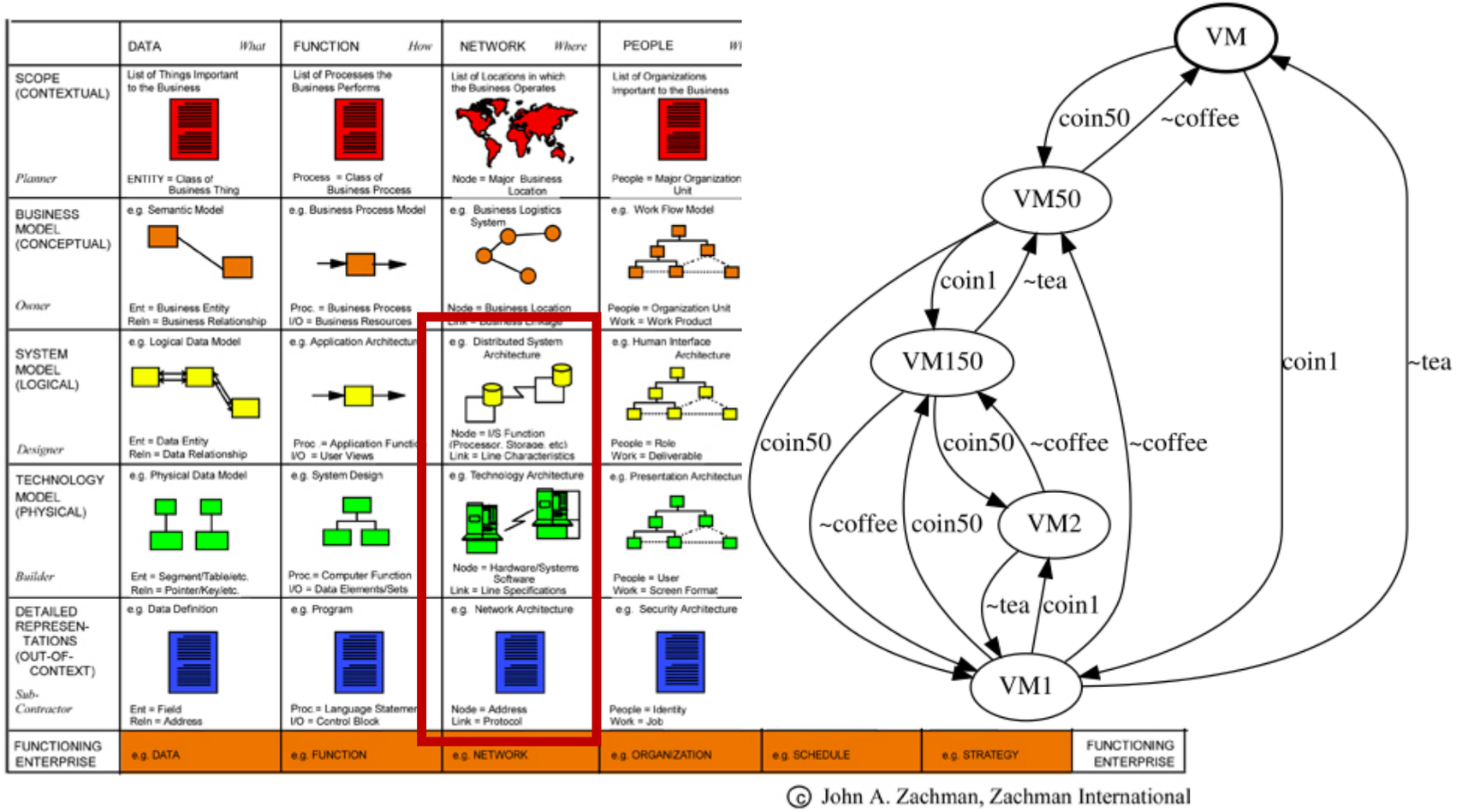
© John A. Zachman, Zachman International

Petri Nets (week 8)

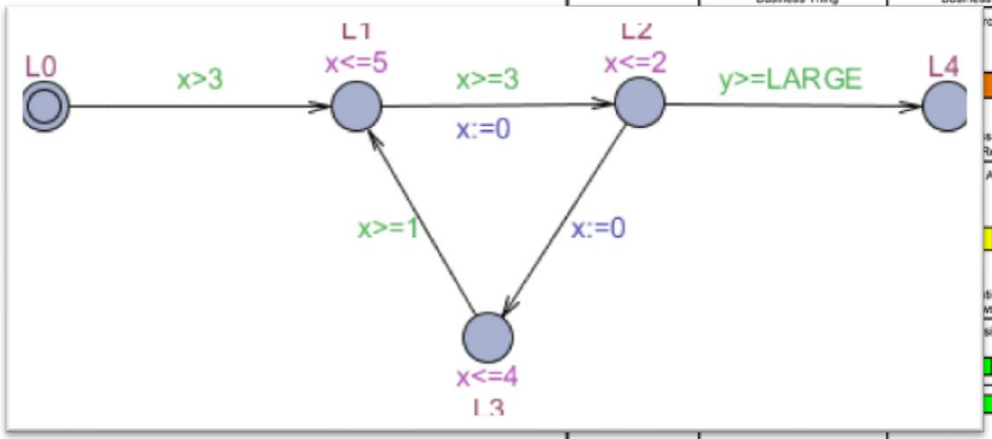


© John A. Zachman, Zachman International

CCS and Pi-calculus (week 9)

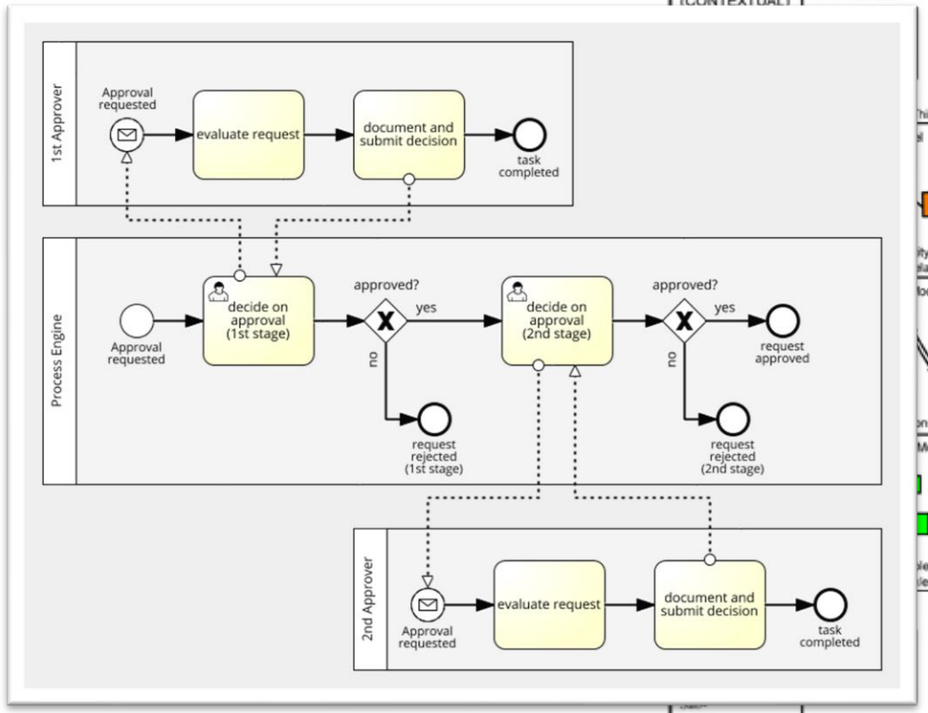


Timed Automata (week 11)



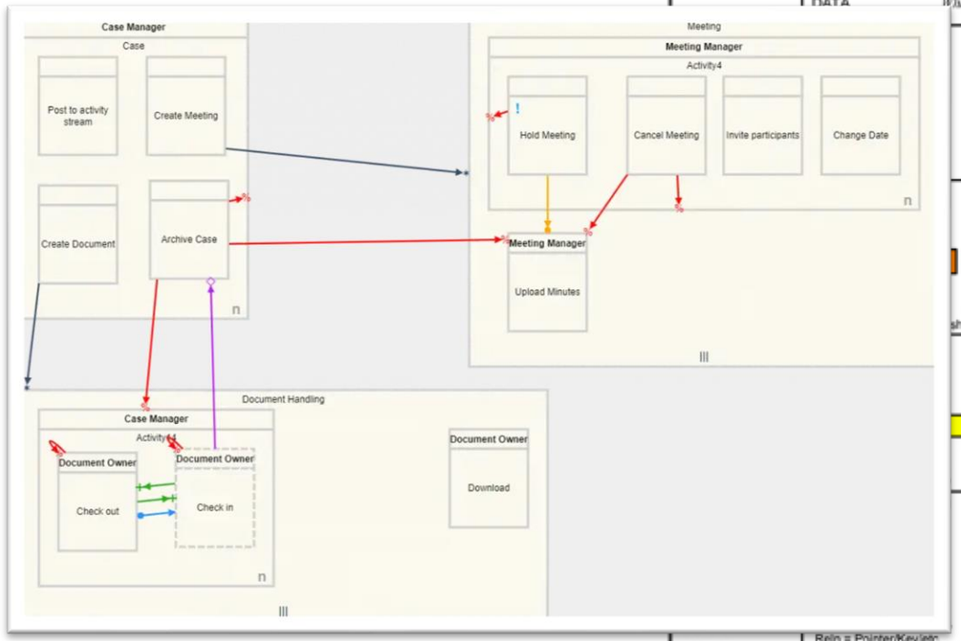
	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL)	List of Things Important to the Business	List of Processes the Business Performs	List of Locations in which the Business Operates	List of Organizations Important to the Business	List of Events/Cycles Significant to the Business	List of Business Goals/Strategies	SCOPE (CONTEXTUAL)
Planner	ENTITY = Class of Business Thing	Process = Class of Business Process	Node = Major Business Location	People = Major Organization Unit	Time = Major Business Event/Cycle	Ends/Means = Major Business Goal/Strategy	*Planner*
	...rocess Model	e.g. Business Logistics System	e.g. Work Flow Model	e.g. Master Schedule	e.g. Business Plan	BUSINESS MODEL (CONCEPTUAL)	
	...s Process ...Resources	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event ...Business Cycle	End = Business Objective ...eans = Business Strategy	*Owner*	
	...Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Processing Structure	e.g. Business Rule Model	SYSTEM MODEL (LOGICAL)	
	...tion Function ...ws	Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	...nd = Structural Assertion ...eans =Action Assertion	*Designer*	
	...sign	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design	TECHNOLOGY MODEL (PHYSICAL)	
Builder	Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	Proc.= Computer Function I/O = Data Elements/Sets	Node = Hardware/Systems Software Link = Line Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	...nd = Condition ...eans = Action	*Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT)	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Specification	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT)
Sub-Contractor	Ent = Field Reln = Address	Proc.= Language Statement I/O = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	...nd = Sub-condition ...eans = Step	*Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

BPMN (week 12 and 14)



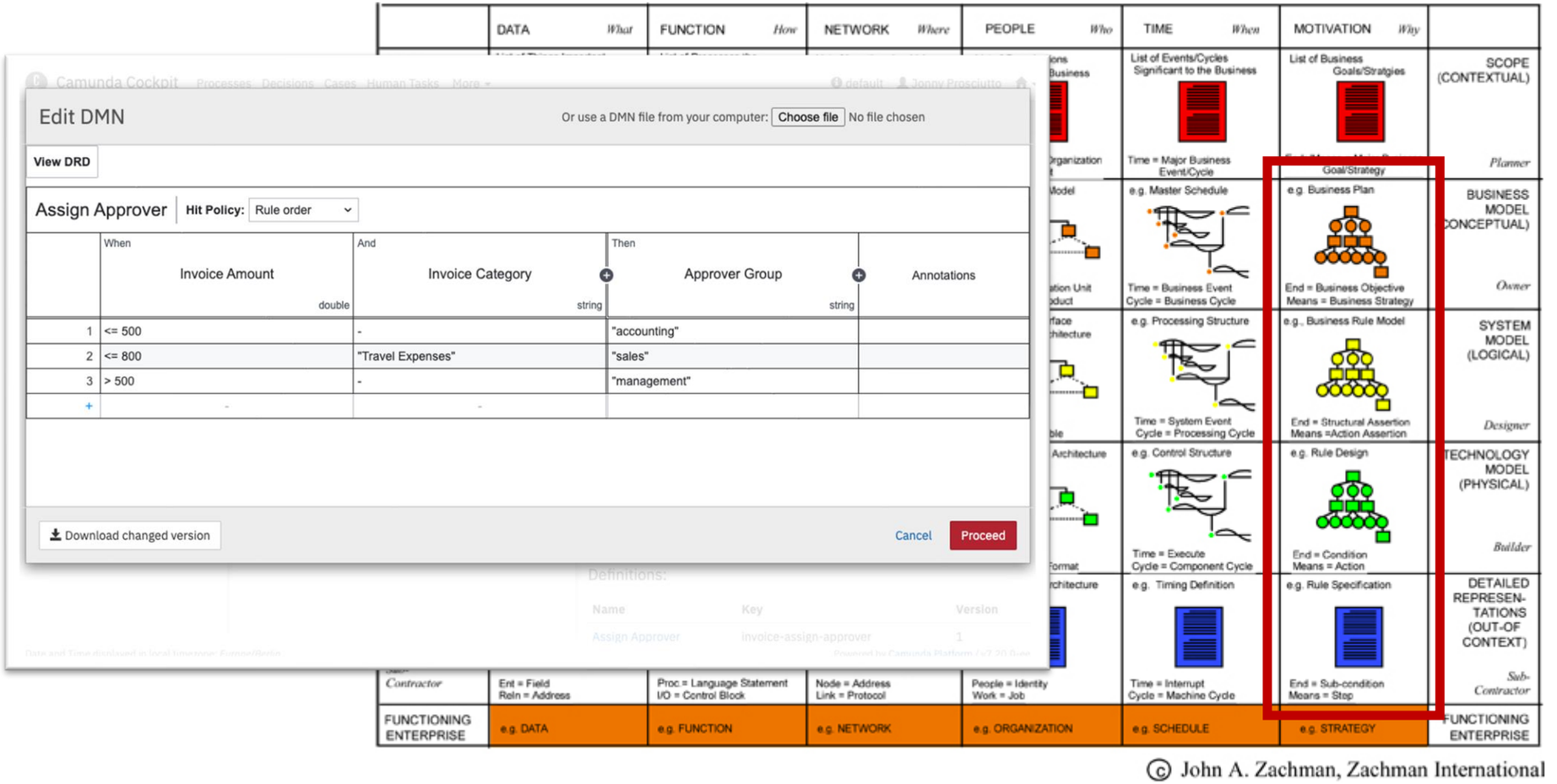
	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL)	List of Things Important to the Business	List of Processes the Business Performs	List of Locations in which the Business Operates	List of Organizations Important to the Business	List of Events/Cycles Significant to the Business	List of Business Goals/Stratgies	SCOPE (CONTEXTUAL) *Planner*
	Ent = Field Reln = Address	Process = Class of	Node = Major Business	People = Major Organization	Time = Major Business	Ends/Means = Major Business Goal/Strategy	
	e.g.	e.g. Business Process Model	e.g. Business Logistics System	e.g. Work Flow Model	e.g. Master Schedule	g. Business Plan	BUSINESS MODEL (CONCEPTUAL) *Owner*
		Proc. = Business Process I/O = Business Resources	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	d = Business Objective eans = Business Strategy	
		e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Processing Structure	, Business Rule Model	SYSTEM MODEL (LOGICAL) *Designer*
		Proc. = Application Function I/O = User Views	Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	nd = Structural Assertion eans =Action Assertion	
		e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	g. Rule Design	TECHNOLOGY MODEL (PHYSICAL) *Builder*
		Proc.= Computer Function I/O = Data Elements/Sets	Node = Hardware/Systems Software Link = Line Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	nd = Condition eans = Action	
		e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	g. Rule Specification	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT) *Sub-Contractor*
	Ent = Field Reln = Address	Proc.= Language Statement I/O = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	nd = Sub-condition eans = Step	
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

DCR Graphs (week 15 and 16)

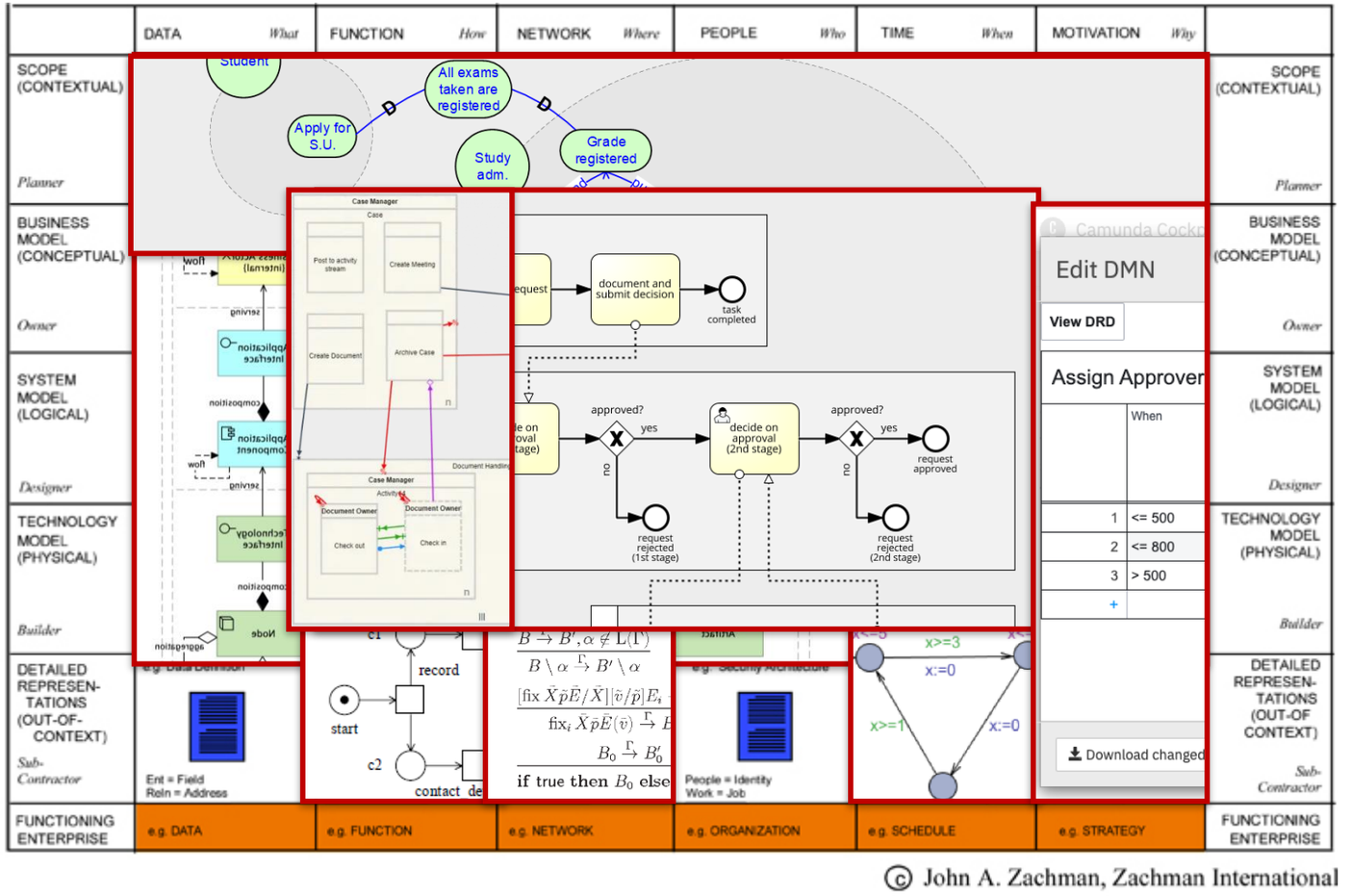


	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
		List of Processes the Business Performs	List of Locations in which the Business Operates	List of Organizations Important to the Business	List of Events/Cycles Significant to the Business	List of Business Goals/Strategies	SCOPE (CONTEXTUAL) *Planner*
		Process = Class of	Node = Major Business	People = Major Organization	Time = Major Business Event/Cycle	Ends/Means = Major Business Goal/Strategy	
		e.g. Business Process Model	e.g. Business Logistics System	e.g. Work Flow Model	e.g. Master Schedule	e.g. Business Plan	BUSINESS MODEL (CONCEPTUAL) *Owner*
		Proc. = Business Process I/O = Business Resources	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	End = Business Objective Means = Business Strategy	
		e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Processing Structure	e.g. Business Rule Model	SYSTEM MODEL (LOGICAL) *Designer*
		Proc. = Application Function I/O = User Views	Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	End = Structural Assertion Means = Action Assertion	
		e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design	TECHNOLOGY MODEL (PHYSICAL) *Builder*
	Reln = Pointer/Key/etc.	Proc. = Computer Function I/O = Data Elements/Sets	Node = Hardware/Systems Software Link = Line Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	End = Condition Means = Action	
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Specification	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT) *Sub-Contractor*
	Ent = Field Reln = Address	Proc. = Language Statement I/O = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Sub-condition Means = Step	
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

DMN (week 17)


Camunda Cockpit
Edit DMN
Or use a DMN file from your computer: Choose file No file chosen
View DRD
Assign Approver
Hit Policy: Rule order
When
Invoice Amount
double
And
Invoice Category
string
Then
Approver Group
string
Annotations
1 <= 500 - "accounting"
2 <= 800 "Travel Expenses" "sales"
3 > 500 - "management"
Download changed version
Cancel
Proceed
Definitions:
Name
Key
Version
Assign Approver
invoice-assign-approver
1
DATA What
FUNCTION How
NETWORK Where
PEOPLE Who
TIME When
MOTIVATION Why
List of Events/Cycles Significant to the Business
List of Business Goals/Stratgies
SCOPE (CONTEXTUAL)
Planner
Time = Major Business Event/Cycle
Goal/Strategy
e.g. Master Schedule
e.g. Business Plan
BUSINESS MODEL (CONCEPTUAL)
Owner
Time = Business Event Cycle = Business Cycle
End = Business Objective Means = Business Strategy
e.g. Processing Structure
e.g., Business Rule Model
SYSTEM MODEL (LOGICAL)
Designer
Time = System Event Cycle = Processing Cycle
End = Structural Assertion Means =Action Assertion
e.g. Control Structure
e.g. Rule Design
TECHNOLOGY MODEL (PHYSICAL)
Builder
Time = Execute Cycle = Component Cycle
End = Condition Means = Action
e.g. Timing Definition
e.g. Rule Specification
DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT)
Sub-Contractor
Time = Interrupt Cycle = Machine Cycle
End = Sub-condition Means = Step
Ent = Field Reln = Address
Proc = Language Statement I/O = Control Block
Node = Address Link = Protocol
People = Identity Work = Job
FUNCTIONING ENTERPRISE
e.g. DATA
e.g. FUNCTION
e.g. NETWORK
e.g. ORGANIZATION
e.g. SCHEDULE
e.g. STRATEGY
FUNCTIONING ENTERPRISE
© John A. Zachman, Zachman International

Overall



© John A. Zachman, Zachman International

Characteristics of languages being taught

- **Models as sketches**: You may use these languages primarily for communication and analysis of specific aspects of the system
  - I*, Automata, CCS and Pi calculus, Petri Nets, Timed Automata, Archimate

- **Models as blueprints**: You may use these languages as representations of existing implementations in a system
  - E.g.: BPMN, DCR graphs

- **Models as programs**: You may use these languages as a low-code approach to system implementation
  - E.g.: BPMN, DCR graphs, DMN