

AUT

COMP 501 – Part Two



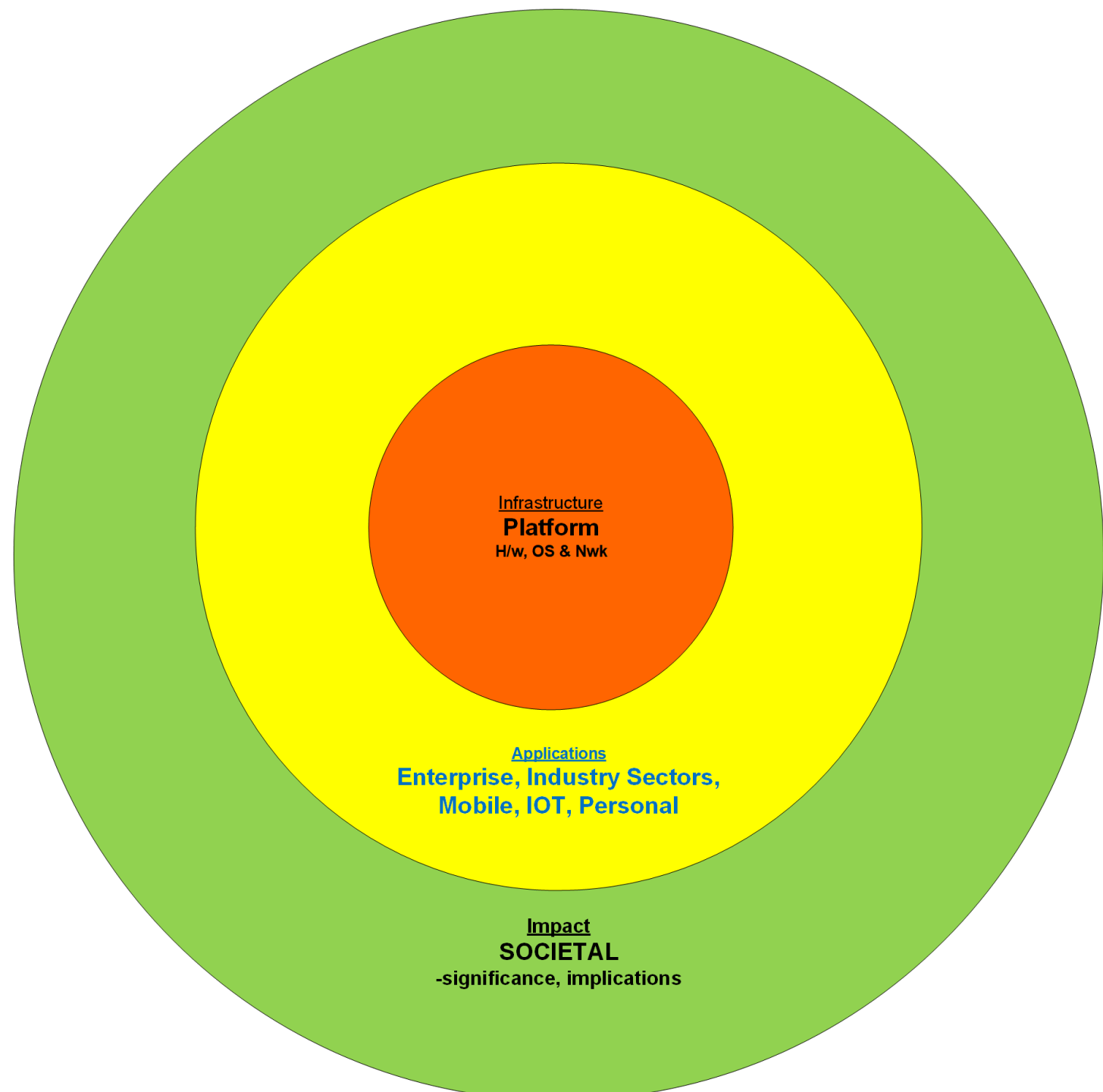
Course lecturer –
Assoc Prof Tony Clear

COMP 501 – Part B Requirements and Modelling

Course lecturer – Assoc Prof Tony Clear

Computing Technology in Society

Paper Overview





Systems Analysis and Design

System Development

- **Systems Analysis and Design** *[may have other names]*
 - Step-by-step process for developing high-quality information systems
 - **Information systems:** Combination of technology, people, and data to perform certain business functions
- **What Does a 'Systems Analyst' Do?** *[may have many other names]*
 - Plans, develops, and maintains information systems
 - Manages IT projects, including tasks, resources, schedules, and costs
 - Conducts meetings, delivers presentations, and writes memos, reports, and documentation

Systems Analysis and Design

- Step-by-step process for developing high-quality Enterprise Systems (ES)
- Who is responsible?
 - Organizations may have an IT department
 - IT department may have a software engineering unit
 - Software engineering units consist of development team (s) (systems analysts/designers/developers), a quality assurance team (testers) and a documentation team (technical writers).
 - Systems Analysts focus on the analysis/design work
 - They also plan, help develop/test, and maintain information systems

Techniques to Understand the Business to Design the System

Business Process modeling (BPM)

- BPM is carried out by business analysts and managers
- BPM is the activity of representing processes (operations) & information needs of a business
- Business process modeling notation (BPMN)

Business Profile

- Mission statement
- **Organizational structure**
- Business processes
- Strengths Markets
- IT infrastructure
- Customers

Business Models

- Business model - graphically displays one or more **business process**
- **Business process** - is a specific set of transactions, events and results that can be described and documented

Systems Development Skills: Systems Analyst

- Investigates, analyzes, designs, develops, installs, evaluates, and maintains a company's information systems
- Constantly interacts with users and managers within and outside the organization
- **Knowledge, Skills, and Education**
 - Technical knowledge
 - Communication and business skills
 - **Critical thinking skills**
 - Education - A college degree in information systems, science, or business
 - Some IT experience is required
 - Certification
 - Helps IT professionals learn new skills and gain recognition for their efforts

Systems Development Skills: Systems Analyst

Technically qualified

- Degree in IT, CS, SE or IS
- Certification course

Business oriented

- Understand about business processes or operations, customers, organizational structures, values, culture
- Know more about the IT needs of the business than managers would know themselves
- Provide business solutions (ES) based on business needs rather than the technology trend.

Highly motivated

- You have firm belief and courage
- Have a positive attitude; do job with pride and passion
- Show respect; recognize other people's good qualities or achievements.
- Have empathy & sincerity

Communication skills (teamwork)

- Understand problems
- exchanging information all the time
- build relationships & connections
- communicate negative or difficult messages without creating conflict or destroying trust

Strong analytical and critical thinking skills

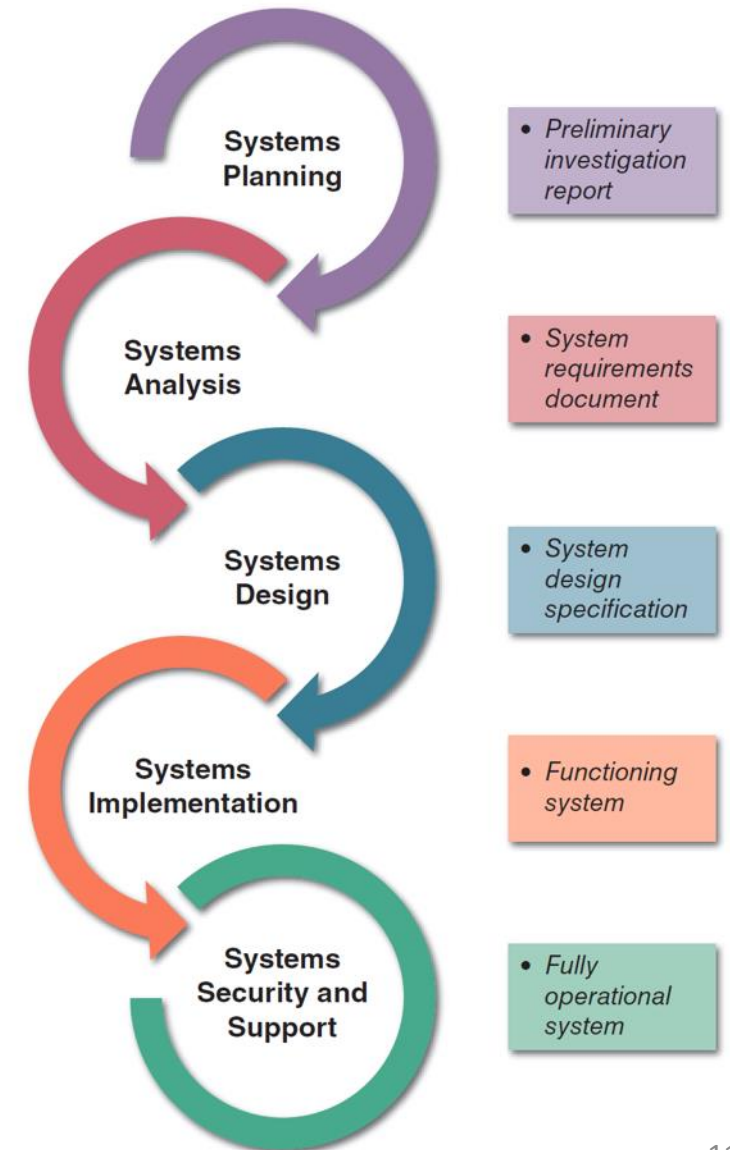
- break a problem down in parts
- provide solution to each part
- Suggest new ideas

Systems Analyst: The Role

- **Role**
 - Acts a translators to managers and programmers
 - A company's best line of defense in an IT disaster
 - Most valuable skill - The ability to listen
 - Seeks feedback from users to ensure that systems do not deviate from accomplishing set objectives
- **Tasks systems analyst does to help develop Enterprise Systems**
 - Translate business requirements into IT projects
 - Plan projects; develop schedules & estimate's cost
 - Document business profile, (review/document business processes), create models (DFDs)
 - Help to create various designs
 - Select hardware & select software packages
 - Conduct meetings, deliver presentations, write memos, produce reports and other documentation in regards to project
 - Test
 - Train users

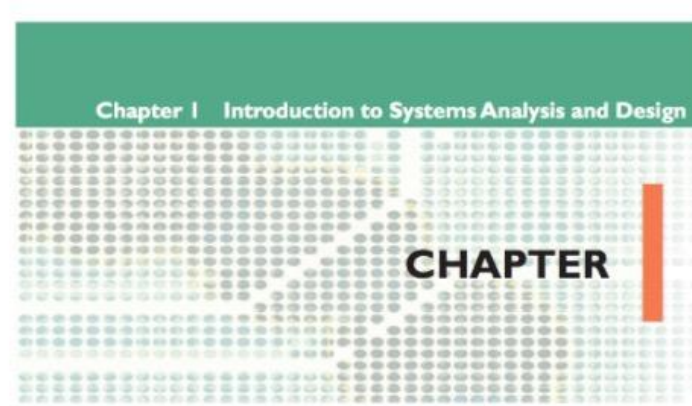
Systems Development Method for Structured Analysis

- The SDLC model usually includes five steps
 - Systems planning
 - Systems analysis
 - Systems design
 - Systems implementation
 - Systems support and security



Related Chapter in eBook

Table Of Contents	
	CP1117: INFS500 Enterprise Systems (Custom) Tiller
ii	Title page
iii	Copyright page
iv	Table of Contents
1	Extract 1 Introduction to Systems Analysis and Design
40	Extract 2 Analysing the Business Case
70	Extract 3 Data and Process Modeling
106	Extract 4 Development Strategies
138	Extract 5 Managing Systems Implementation
186	Extract 6 Managing Systems Support and Security
239	Extract 7 Enterprise Systems
277	Extract 8 Information and Decision Support Systems
331	Extract 9 Knowledge Management and Specialised...



Introduction to Systems Analysis and Design

Chapter 1 is the first of three chapters in the systems planning phase. This chapter describes the role of information technology in today's dynamic business environment. This chapter describes the development of information systems, systems analysis and design concepts, and various systems development methods. This chapter also describes the role of the information technology department and its people.

The chapter includes four "Case in Point" discussion questions to help contextualize the concepts described in the text. The "Question of Ethics" invites examination of the ACM's code of ethics and those of a developing systems analyst.

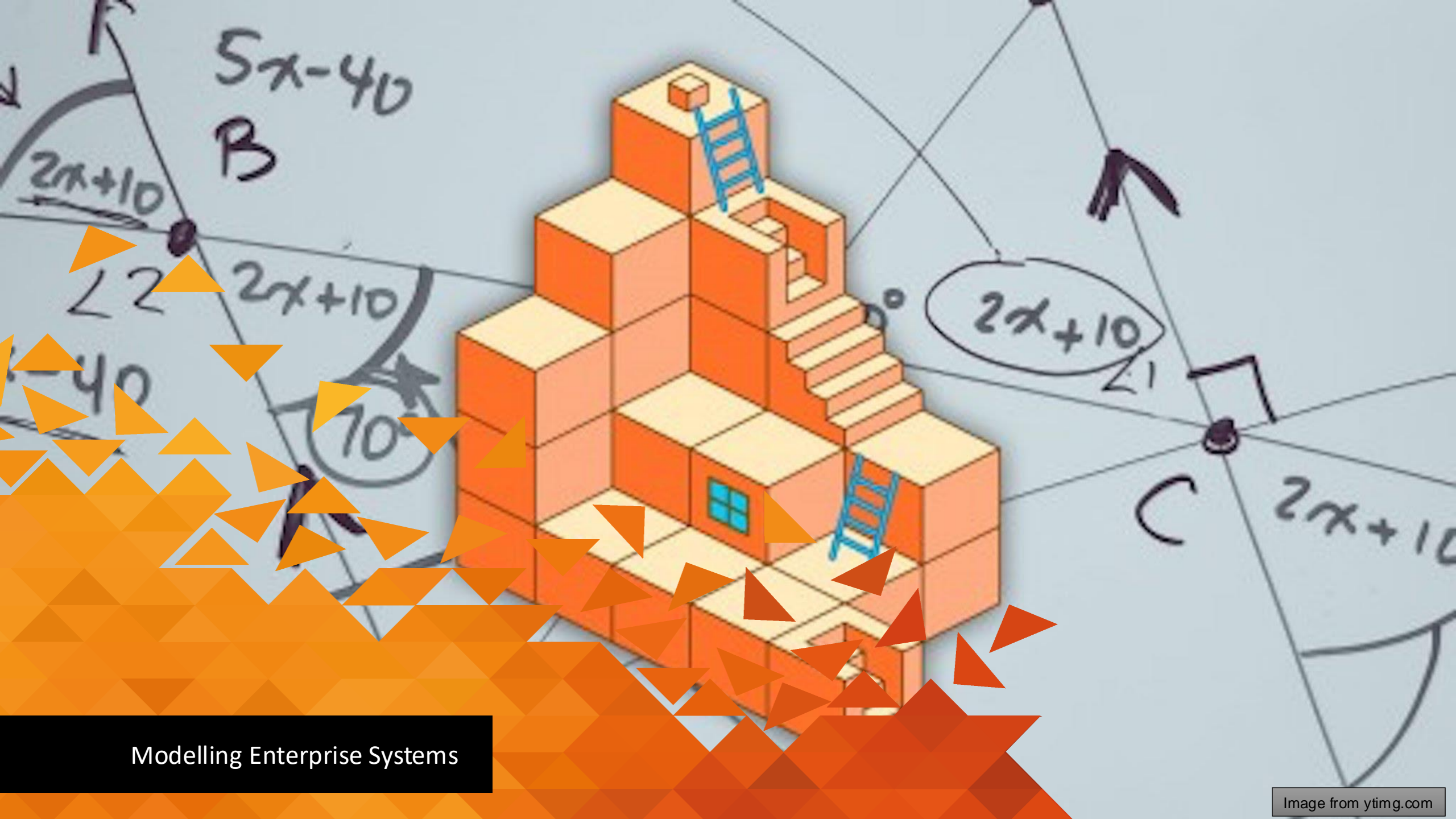
LEARNING OBJECTIVES

When you finish this chapter, you should be able to:

- Describe the impact of information technology
- Define systems analysis and design and the role of a systems analyst
- Define an information system and describe its components
- Explain how to use business profiles and models
- Explain Internet business strategies and relationships, including B2C and B2B
- Identify various types of information systems and explain who uses them
- Distinguish among structured analysis, object-oriented analysis, and agile methods

CHAPTER CONTENTS

- 1.1 Introduction
- 1.2 What Is Information Technology?
Case in Point 1.1: Cloud Nine Financial Advisors
- 1.3 Information System Components
- 1.4 Business Today
- 1.5 Modeling Business Operations
- 1.6 Business Information Systems
- 1.7 What Information Do Users Need?
- 1.8 Systems Development Tools
- 1.9 Systems Development Methods
- 1.10 The Information Technology Department
Case in Point 1.2: Global Hotels and Momma's Motels
Case in Point 1.3: What Should Lisa



Modelling Enterprise Systems

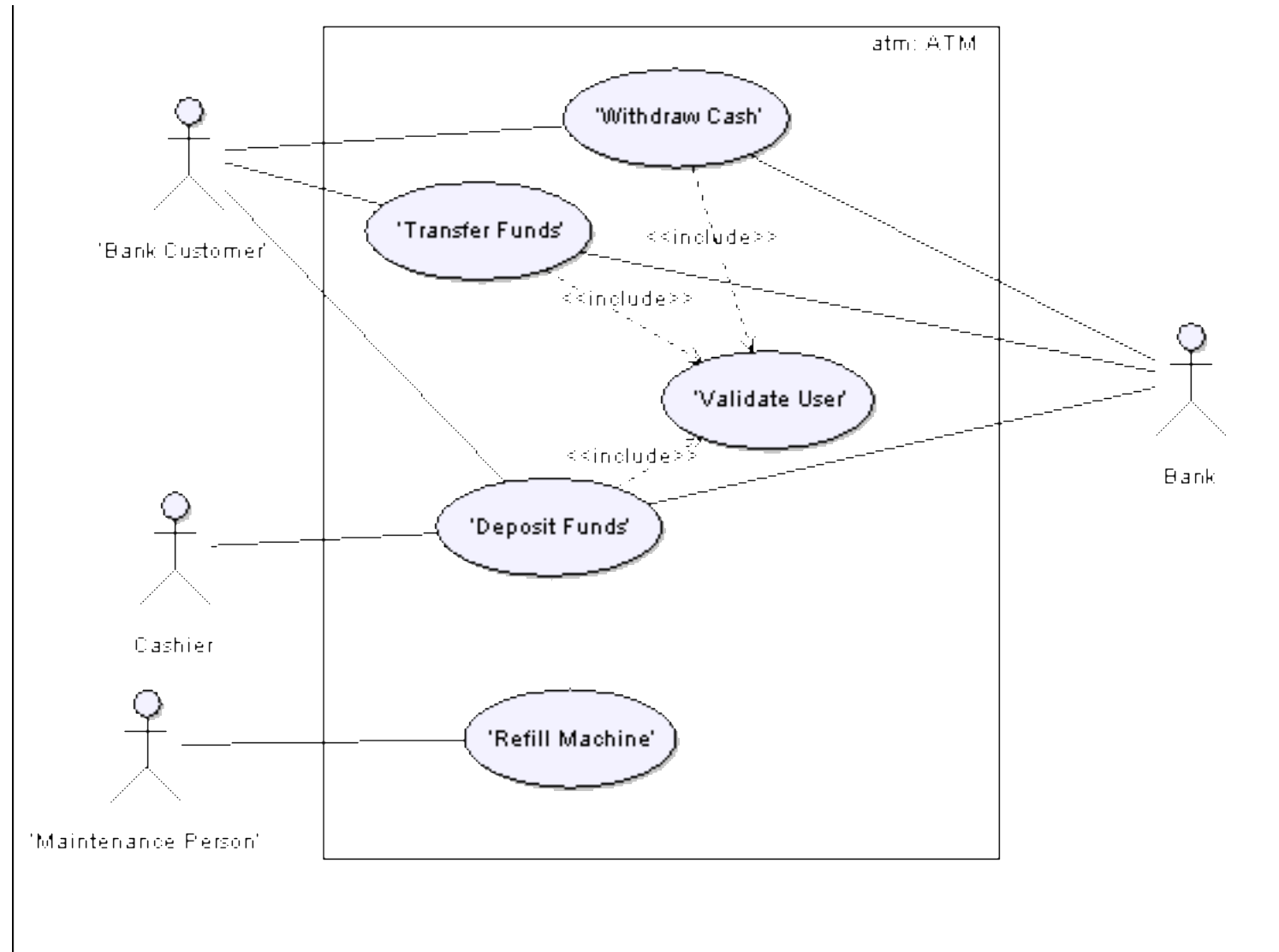
Modelling

- Analysts and Developers will use various models to learn, get feedback, and communicate what needs to be implemented and how it needs to be implemented
 - Requirements model
 - Describes the features that a system must provide,
 - Data Flow Diagrams (DFD)
 - Object model (O/O analysis/design)
 - Describes objects, which combine data and processes
 - Data model
 - Describes data structures and design
 - Network model
 - Describes the design and protocols of telecommunications links
 - Process model
 - Describes the logic of a process that programmers use to write code.

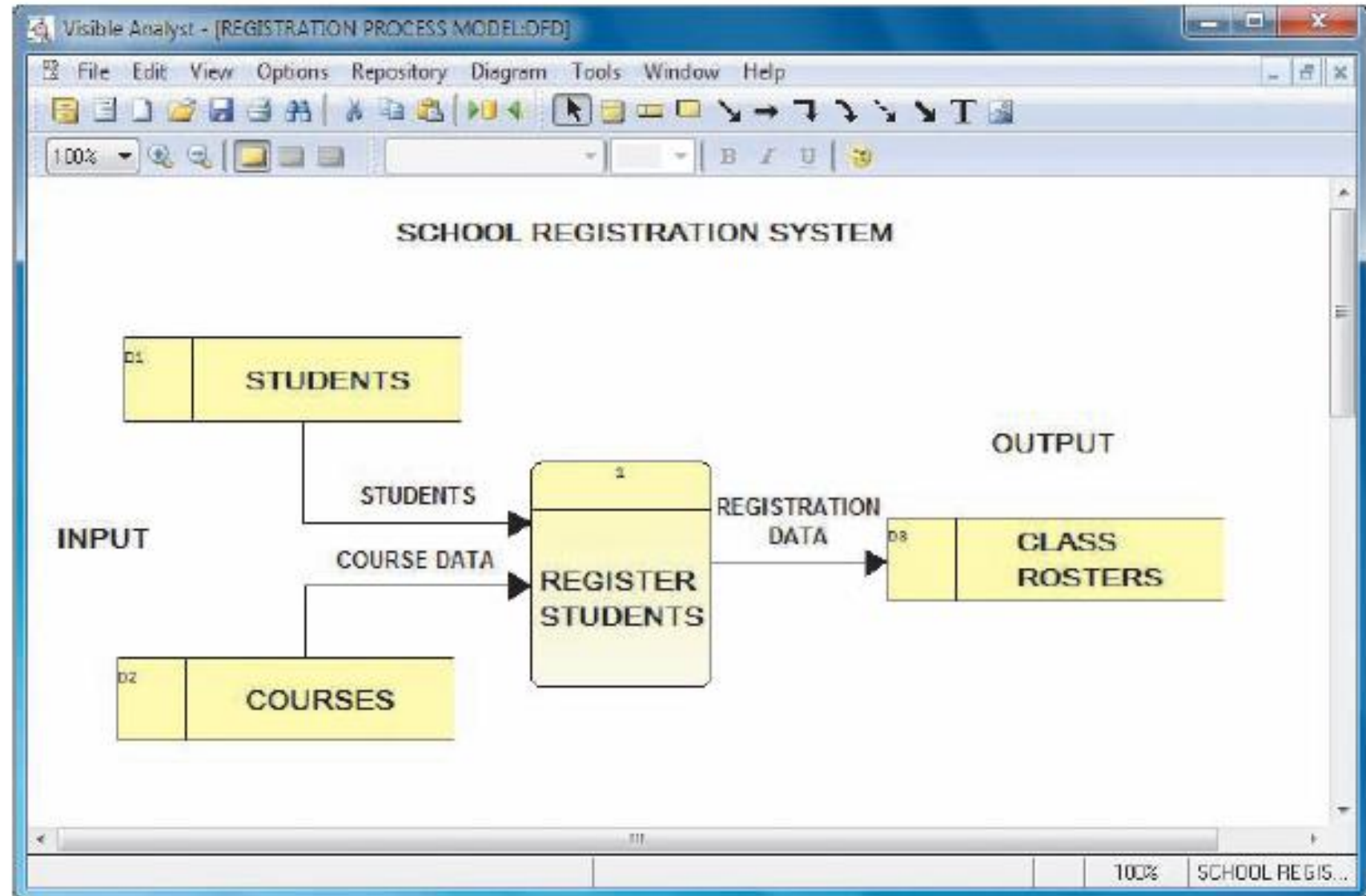


We will learn DFDs in labs

Use case diagram / model



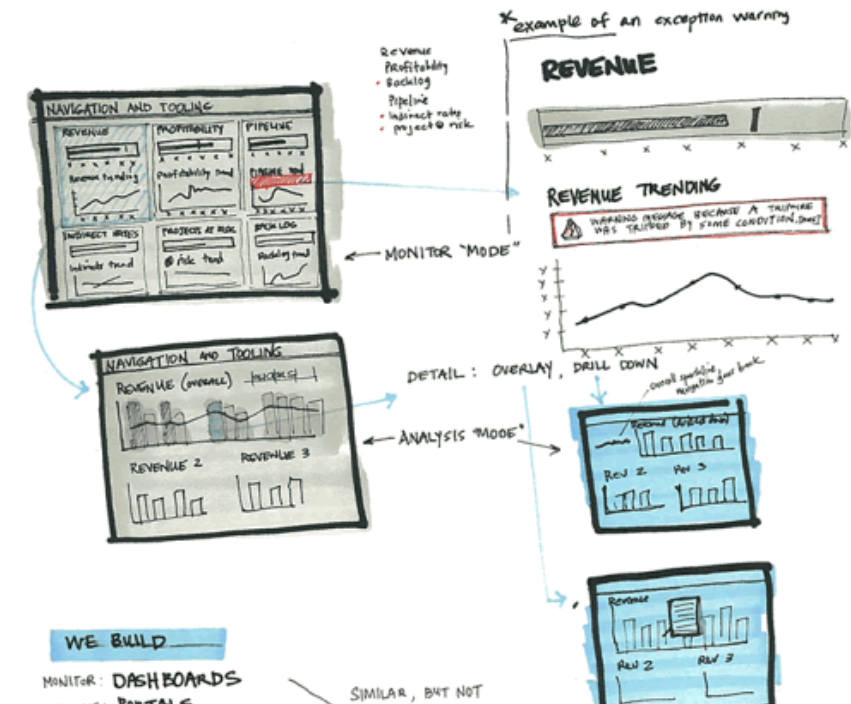
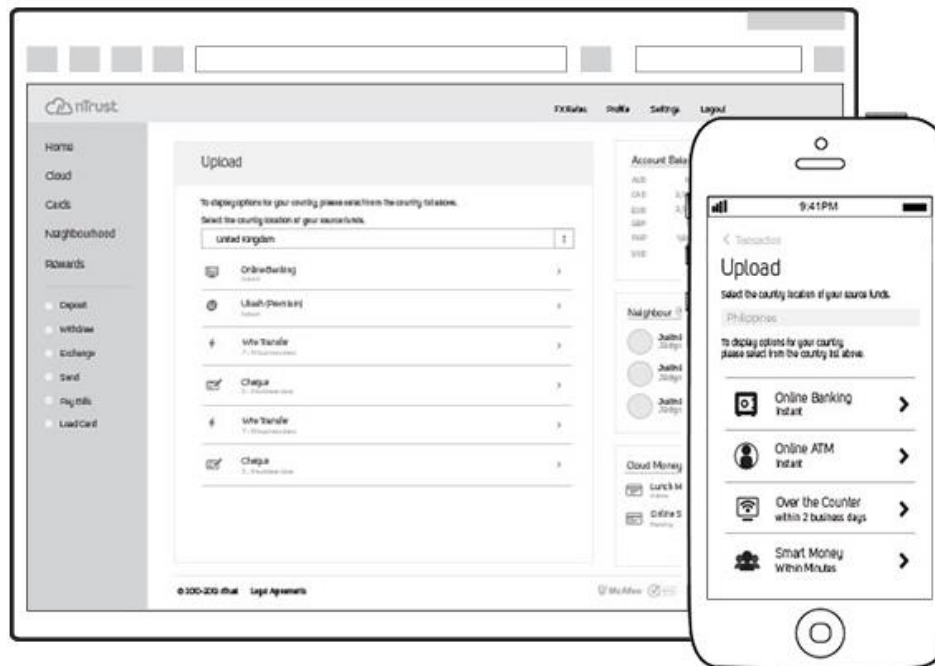
Data flow model example



Systems Development Technique

- Prototyping

- Allows to identify, test, and get a feedback on features
- It is a working version of a system but not fully functional system
- Test input, output and user interface before making a final implementation decision



Systems Development Tools

- A confusing plethora of tools to help automate parts of the development lifecycle

<https://digital.ai/sites/default/files/pictures/2020-06/>

Digital.ai_Periodic-Table-of-DevOps.pdf

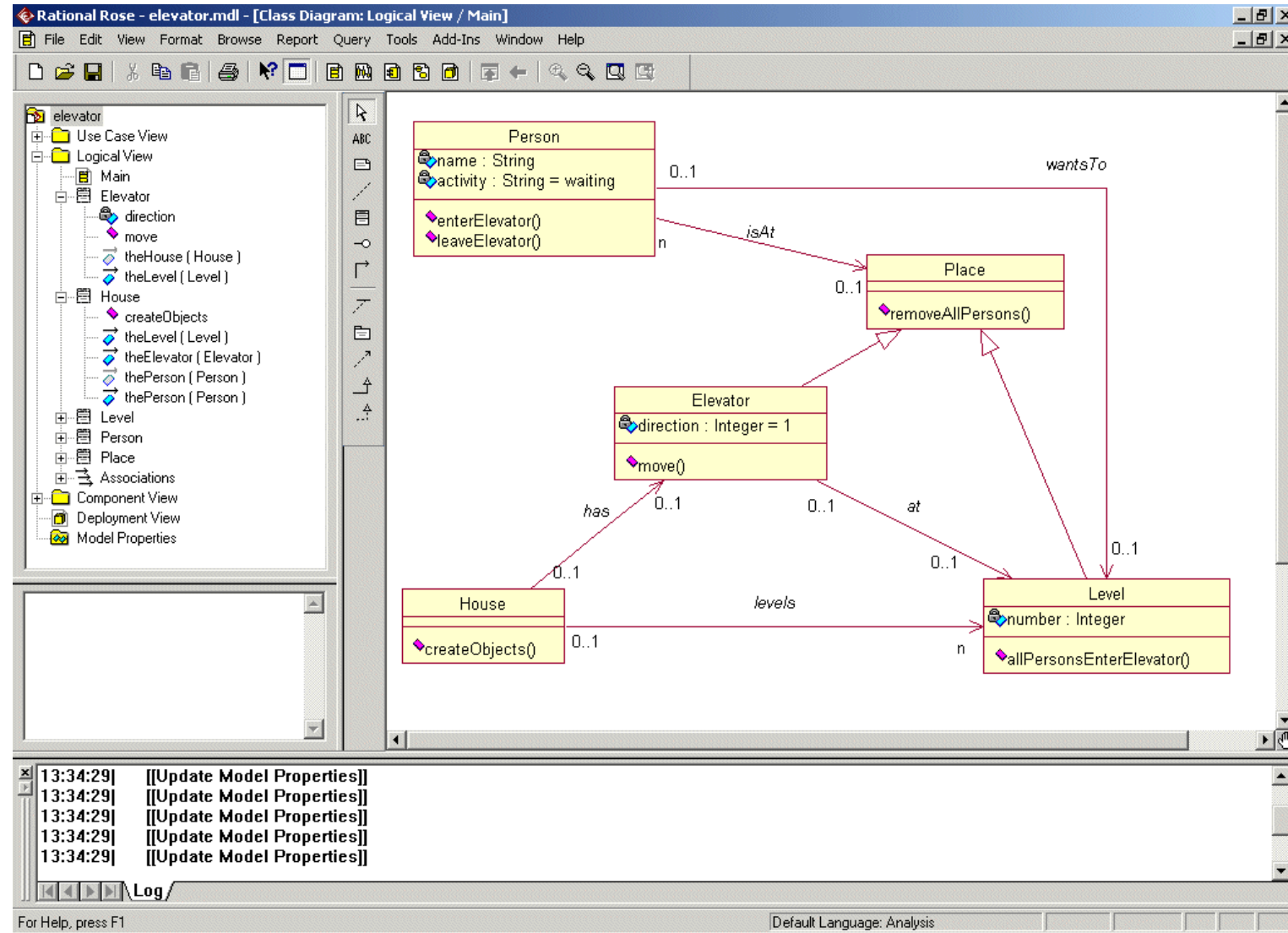
PERIODIC TABLE OF DEVOPS TOOLS (V2)										EMBED DOWNLOAD ADD			Digital.ai - P...																						
<div><div>Os Open Source</div><div>Fr Free</div><div>Fm Freemium</div><div>Pd Paid</div><div>En Enterprise</div></div> <div><div>SCM</div><div>CI</div><div>Deployment</div><div>Cloud / Iaas / Pass</div><div>BI / Monitoring</div></div> <div><div>Database Mgmt</div><div>Repo Mgmt</div><div>Config / Provisioning</div><div>Release Mgmt</div><div>Logging</div></div> <div><div>Build</div><div>Testing</div><div>Containerization</div><div>Collaboration</div><div>Security</div></div>																Aws AmazonWeb Services																			
1 Fm Gh Github		3 Os Gt Git		4 Pd Dm DBmaestro								5 En Ch Chef		6 En Pu Puppet		7 Os An Ansible		8 En Sl Salt		9 Os Dk Docker		10 Pd Az Azure													
11 Fm Bb Bitbucket		12 Os Lb Liquibase								13 Os Ot Otto		14 En Bl BladeLogic		15 Os Va Vagrant		16 Fr Tf Terraform		17 Os Rk rkt		18 En Gc Google Cloud Platform															
19 Os Gl GitLab		20 En Rg Redgate		21 Os Mv Maven		22 Os Gr Gradle		23 Os At ANT		24 Os Fn FitNesse		25 Fr Se Selenium		26 Os Ga Gatling		27 Fr Dh Docker Hub		28 Os Jn Jenkins		29 Pd Ba Bamboo		30 Os Tr Travis CI		31 Pd Gd Deployment Manager		32 Os Sf SmartFrog		33 Os Cn Consul		34 Os Bc Bcf92		35 Os Mo Mesos		36 En Rs Rackspace	
37 Os Sv Subversion		38 En Dt Datacal		39 Os Gt Grunt		40 Os Gp Gulp		41 Os Br Broccoli		42 Fr Cu Cucumber		43 Os Cj Cucumber.js		44 Fr Qu Qunit		45 Os Npm npm		46 Fm Cs Codeship		47 Pd Vs Visual Studio		48 Fm Cr CircleCI		49 Fr Cp Capistrano		50 Fr Ju JuJu		51 Os Rd Rundeck		52 Os Cf CFEngine		53 Fr Ds Swarm		54 Os Op OpenStack	
55 Os Hg Mercurial		56 En Dp Delphix		57 Fr Sb sbt		58 Os Mk Make		59 Os Ck CMake		60 Fr Ju JUnit		61 Fr Jm JMeter		62 Fr Tn TestNG		63 Os Ay Artifactory		64 Fm Tc TeamCity		65 Fm Sh Shippable		66 Os Cc CruiseControl		67 En Ry RapidDeploy		68 Fm Cy CodeDeploy		69 En Oc Octopus Deploy		70 En No CA Nolio		71 Os Kb Kubemetes		72 Fm Hr Heroku	
73 En Cw ISPW		74 En Id Idera		75 Os Msb MSBuild		76 Os Rk Rake		77 Fr Pk Packer		78 Os Mc Mocha		79 En Xltv XL TestView		80 Os Jm Jasmine		81 Os Nx Nexus		82 Os Co Continuum		83 Fm Ca Continua CI		84 Pd So Solano CI		85 En Xld XL Deploy		86 En EB ElectricBox		87 Fm Dp Deploybot		88 En Ud UrbanCode Deploy		89 Os Nm Nomad		90 En Os OpenShift	

XebiaLabs
Deliver Faster

Follow @xebialabs

91 En Xlr XL Release	92 En Ur UrbanCode Release	93 En Bm BMC Release Process	94 En Hp HP Codar	95 En Au Automic	96 En Pl Pivotal Release	97 En Sr Serena Release	98 Pd Tfs Team Foundation	99 Fm Tr Trello	100 Pd Jr Jira	101 Fm Rf HipChat	102 Fm Sl Slack	103 Fm Fd Flowdock	104 Pd Pv Pivotal Tracker	105 En Sn ServiceNow
106 Os Ki Kibana	107 Fm Nr New Relic	108 Os Ni Nagios	109 Os Zb Zabbix	110 En Dd Datalog	111 Os El Elasticsearch	112 Os Ss StackState	113 En Sp Splunk	114 Fm Le Logentries	115 Fm Sl Sumo Logic	116 Os Ls Logstash	117 Os Gr Graylog	118 Os Sn Snort	119 Os Tr Tripwire	120 En Ff Fortify

CASE tool – Rational Rose





Agile Models Distilled: Potential Artifacts for Agile Modeling

[Home](#)

[Start Here](#)

[Core Practices](#)

[Disciplines](#)

[Artifacts](#)

[Resources](#)

[Contact Me](#)

Search

Choose Your WoW!

To be effective, the principle **Multiple Models** tells us that agile modelers should know a wide variety of modeling techniques so that they have the skills and knowledge to **apply the right artifact(s)** for the situation at hand. Unfortunately this is easier said than done. This page links to summary descriptions of a wide variety of modeling artifacts. Each page describes the artifact, provides an example or two, and provides links to suggested resources.

Some, but not all, of the potential models that you may want to create on a software development project include:

AGILE Modelling – Potential Models

- Acceptance Test
- Business Rule (Template)
- Change Case (Template)
- Class Responsibility Collaborator (CRC) model
- Constraint
- Contract model (Template)
- Data Flow Diagram (DFD)
- Domain Model
- Essential/Abstract Use Case (Template)
- Essential/Abstract User Interface Prototype
- Feature
- Free-Form Diagrams
- Flow Chart
- Glossary
- Logical Data Model (LDM)
- Mind Map
- Network Diagram
- Object Role Model (ORM) Diagram
- Personas
- Physical Data Model (PDM)
- Robustness Diagram
- Security Threat Model
- System Use Case (Template)
- Technical Requirement
- UML Activity Diagram
- UML Class Diagram



We will learn DFDs in labs

<http://agilemodeling.com/artifacts/>

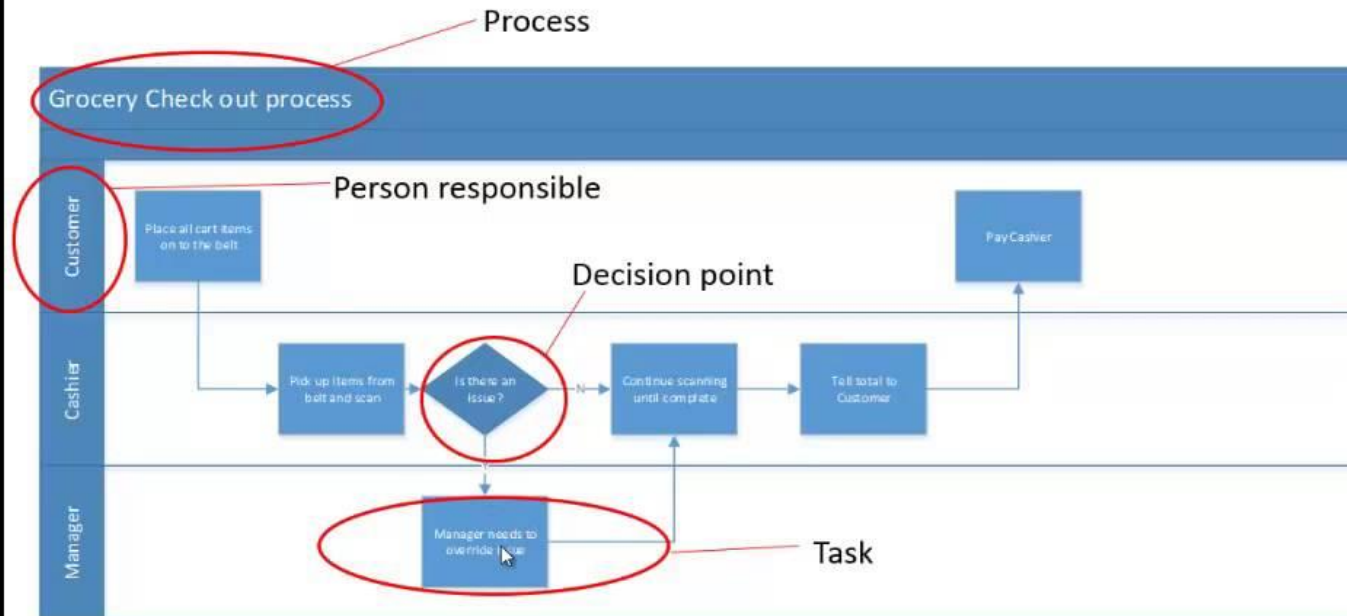
Tutorial Session 6: Modelling Dataflows

1. Modelling dataflows and why?
2. Blog Case – Digital Government: Remodelling a Process
The Service: Applying for NZ Superannuation
<https://www.digital.govt.nz/blog/user-focussed-content-equals-good-business/>
3. DFD's

Modelling Dataflows – Swimlane Process Maps

<https://www.youtube.com/watch?v=wQxnzLu7TqU>

Swim Lane Process Map



Modelling Dataflows – Swimlane Process Maps

1. Modelling Current and Future States

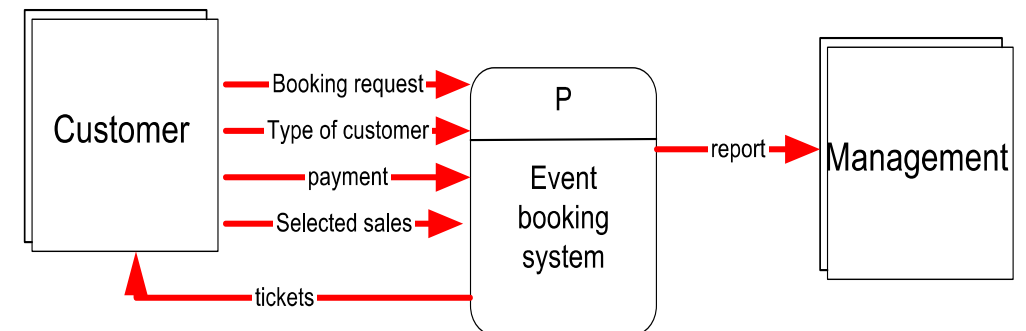
-
- Current State Modelling
- <https://www.youtube.com/watch?v=ImjvmcuhaKY>
-
- Future State Modelling and Process Improvement Design
- <https://www.youtube.com/watch?v=cAkPmSmK6ZM>



Data Flow Diagrams (DFDs)

Objectives of this section

- Describe data and process modeling concepts and tools, including data flow diagrams, and process descriptions
- Describe the symbols used in data flow diagrams and explain the rules for their use
- Exposure to drawing **context diagram**



Process Modelling using DFDs

- DFDs show how the system transforms input data into useful information
- Technique for organising and documenting a system's **processes, inputs, outputs** and **data stores (storage)**
- Also includes the **external entities**.
- Process modelling is structured analysis tool which deals with a business process from the systems owners' and systems users' point of view
- What the system does or must do
- We use DFD to capture system's components (features) and external entities

Process modeling: Data Flow Diagrams (DFD)

- A data flow diagram (DFD) shows how **data moves** through an information system but does **not show program logic or processing steps**
- **Levels of modelling**
 - Context Diagram
 - High level (little detail, completeness of structure)
 - Low level (more detail, decomposed to smaller parts)



System Outline

- The first step towards identifying the components for the process driven model is to complete a System Outline.
- A System Outline identifies the system **inputs, outputs, processes, files (data storage) and external entities**. The processes are the events that occur in the system.
- The details from which the system outline can be derived, will be obtained by interviewing the users throughout the analysis and design phases of the **Systems Development Life Cycle**.

System Outline

Sample system outline. A large collection of these will be created to meet the needs of a complex system.

System Outline

Title	System	Document	Name	Sheet
Input			Processes	
Files (Datastores)			Outputs	
External Entities				
Author			Date	

DFD Notations

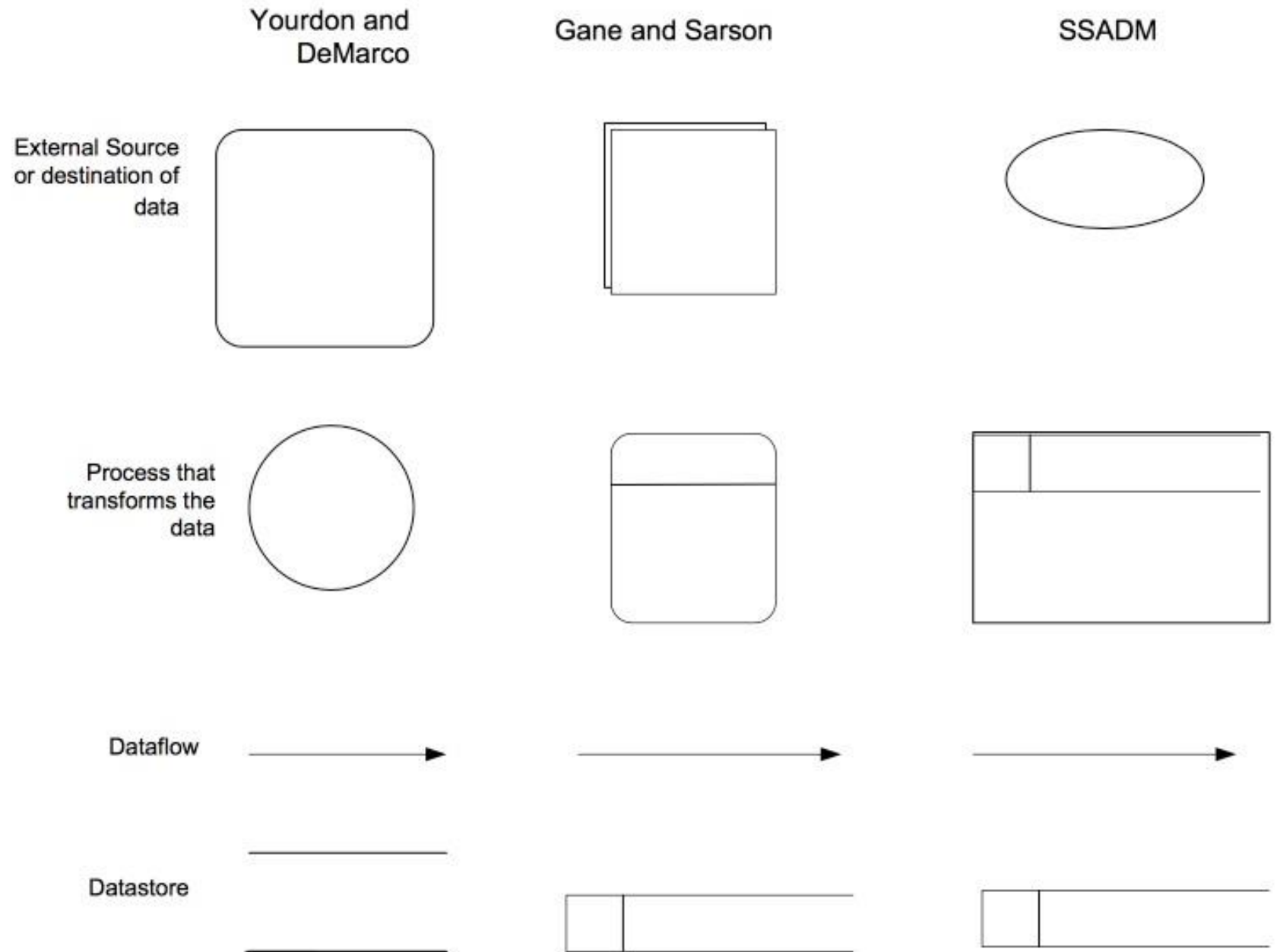
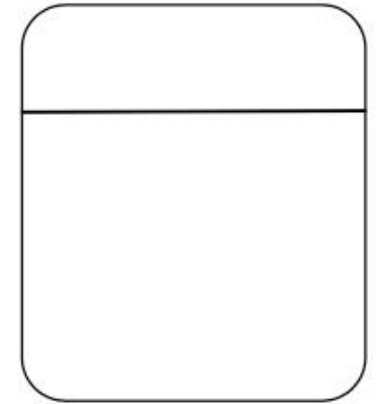


Figure 2.5: Various DFD notations

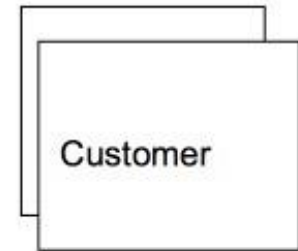
Data Flow Diagrams(DFDs) – Process Symbol

- Drawn as **rectangle with round corners**. Each process must be numbered
- Name of the process starts with a verb followed by a singular object e.g. calculate gross pay, produce invoice, validate customer
- Receives input data and produces output that has a different content, form, or both
- Contain the business logic, also called business rules
- Referred to as a black box



Data Flow Diagrams(DFDs) - External Entities

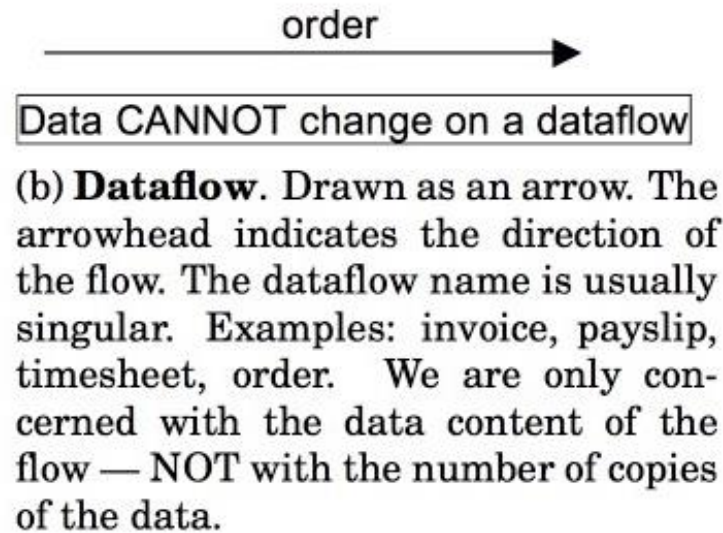
- Provide input or receive output from the system
- **Source** (provides data [input] to the system) or a **Sink** (gets the data [output] from the system)
- Drawn as a **square with another square behind it**
- E.g. Customer, Supplier, IRD, Employee etc.



(a) **External Entity** is a source or sink for data. This is drawn as a square with another square behind it. Examples: Customer, IRD, Employee, Supplier

Data Flow Diagrams(DFDs) – Data Flow

- Drawn as an **arrow**. Arrow head indicates the direction of the flow
- Dataflow name is singular, written on top of the arrow
- E.g. Invoice, payslip, timesheet, order



Data Flow Diagrams(DFDs)

- **Data store (storage)**

- Drawn as an **open-ended rectangle**
- Data is stored for use by a process at a later time
- Data storage also provides input into another process
- **Name of data store is usually plural**
- Each data store is usually numbered for reference purpose e.g. D1
- **Customer master file named as customers, employee master file named as employees**

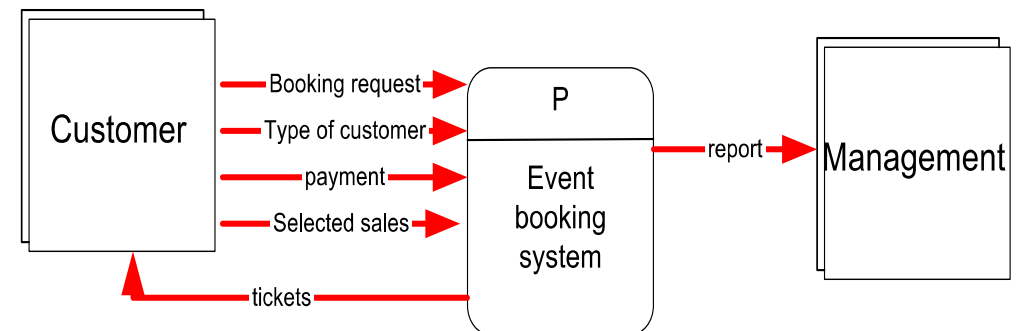
- **Duplicates**

- Sometimes it is necessary to repeat the a symbol on a DFD in order to avoid crossing lines



Drawing context diagram

- Draw the context diagram so that it fits on one page
- Has a **single process** representing the entire system. The name of the process is name of the system
- Identify **all the external entities** to go with the process.
- Shows the system as a single process with external entities
- External entities show all the input and output they provide and receive from the system
- Context diagram helps the analyst to gain an overall view of the system he/she is investigating



DFD Modelling

1. DFD's

Enrolling in the University

<http://agilemodeling.com/artifacts/dataFlowDiagram.htm>

2. Blog Case – Digital Government

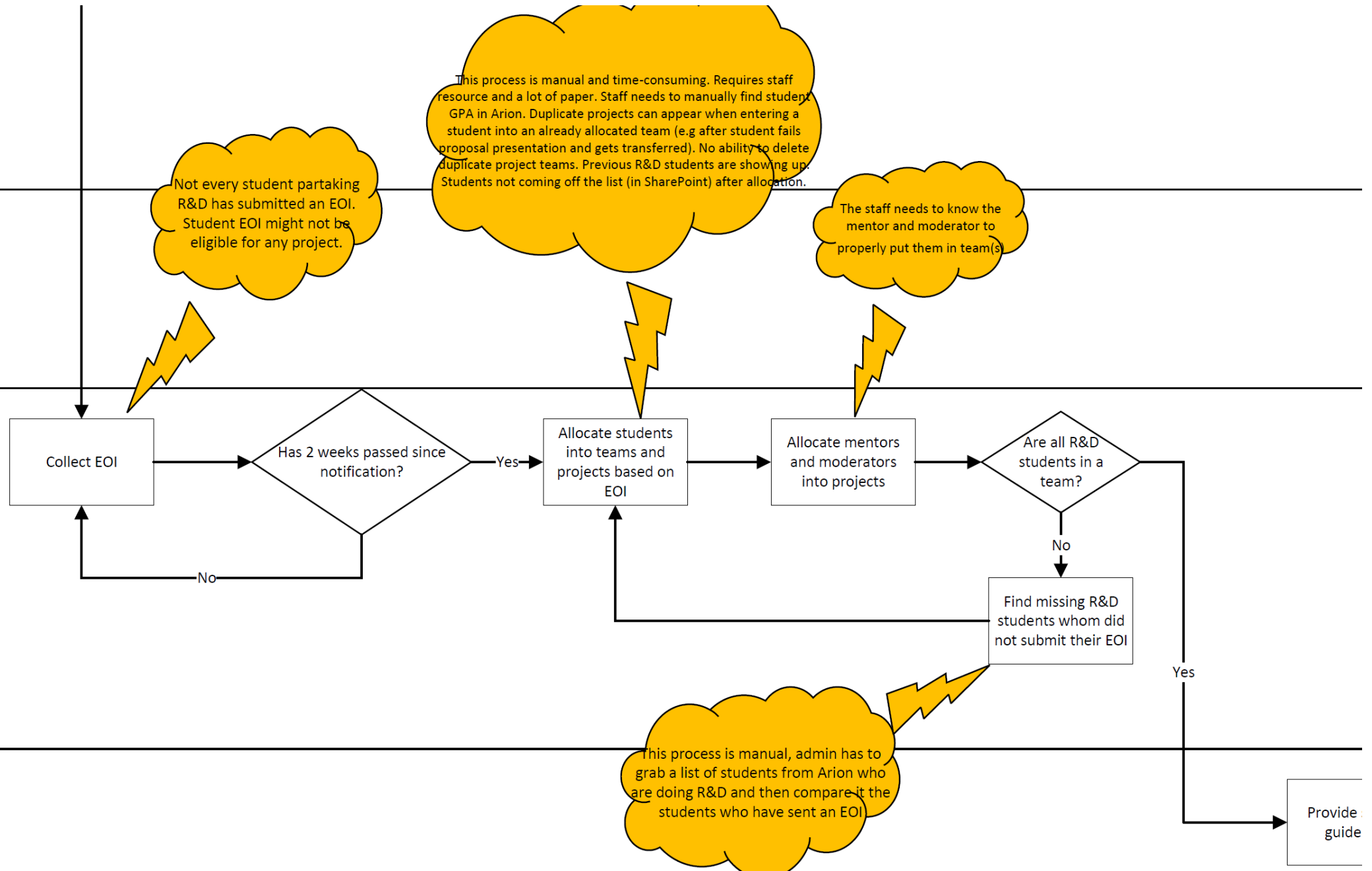
The Service: Applying for NZ Superannuation

<https://www.digital.govt.nz/blog/user-focussed-content-equals-good-business/>

DFD Modelling – BCIS R&D Project - S2/ 2021

1. Consultation Report
2. Pain points – to see next slide
3. Discussion and issues
4. Options and Implications

DFD Modelling – BCIS R&D Project - Pain Points



Tutorial Session 6: Modelling Dataflows (1)

THE STATIONERY STORES SYSTEM

The Stationery Stores System is a case study used by Park Place Training (see References) to teach the general principles of structured analysis and design. For the purpose of illustrating the approach to Mk II function point counting, we will only use a small part of the overall case study. This part concerns the processes from the time when the Stores sees the need to re-order some stationery, through the selection of possible suppliers based on past performance, the preparation of requests for quote, and their issue to suppliers, and the receipt and registration of quotes from suppliers.

In the case study, we are at a point where Data Flow Diagrams ("DFD's") have been produced for the required processes (referred to as a 'First Cut Functional Design'), data entity analysis has been carried out, and the

Tutorial Session 6: Modelling Dataflows (2)

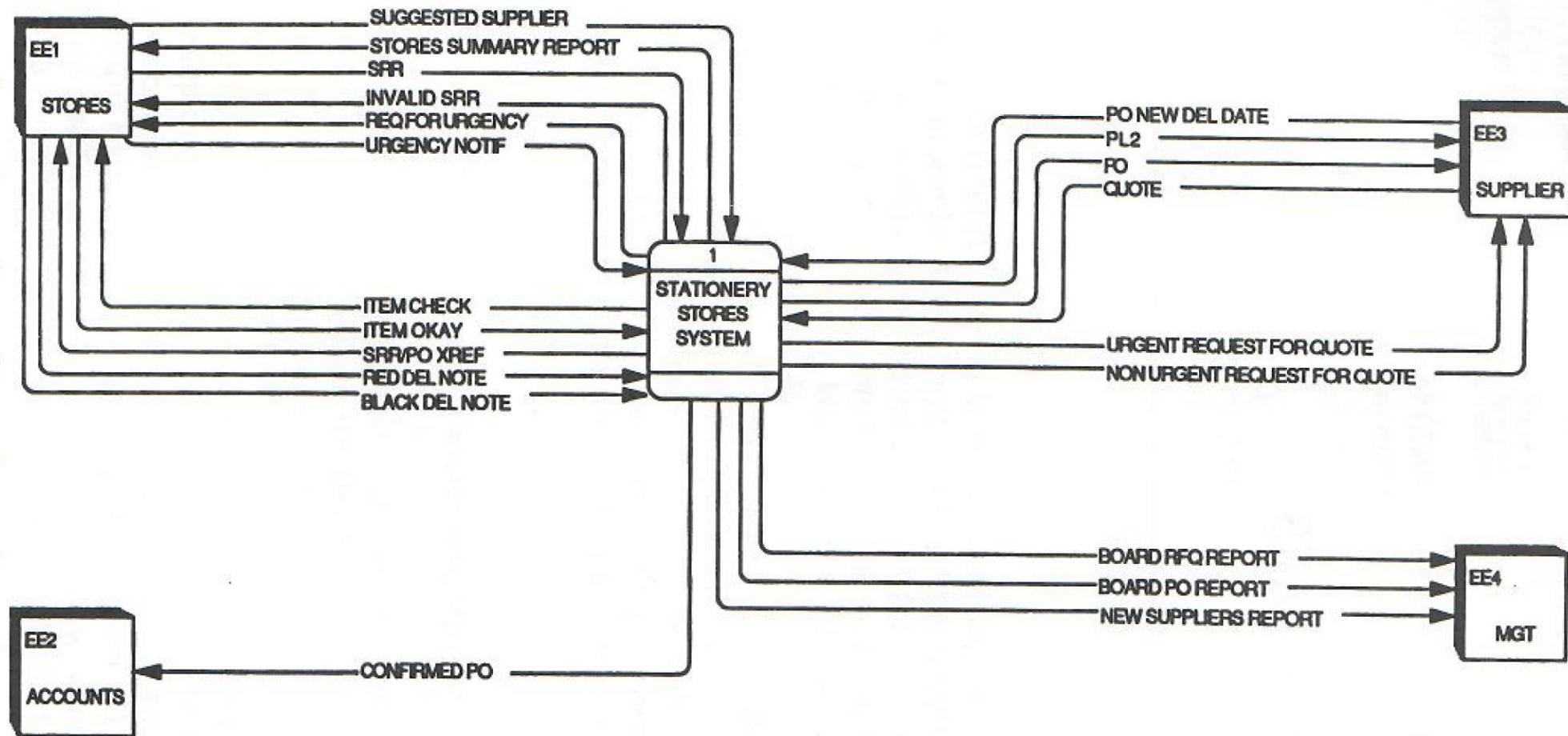


Figure 26 Stationery Stores First Cut Functional Design—Context

Symons, C. R. (1991). *Software sizing and estimating: Mk II FPA (function point analysis)*: John Wiley & Sons, Inc.

Tutorial Session 6: Modelling Dataflows (3)

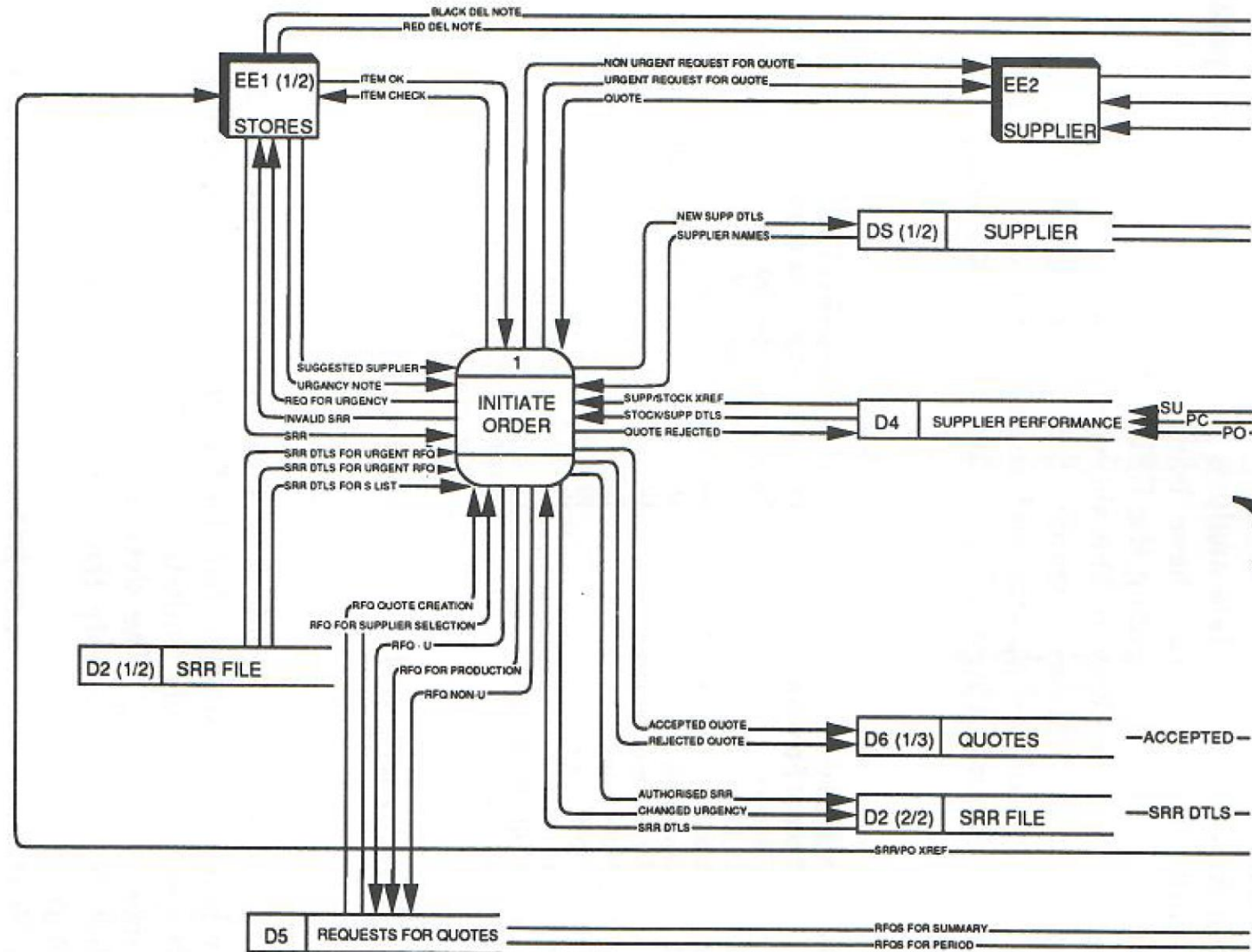


Figure 28(a) First Cut Functional Design—Level 0

Symons, C. R. (1991). *Software sizing and estimating: Mk II FPA (function point analysis)*: John Wiley & Sons, Inc.



Extend your knowledge

1. Watch a video on – what is an Information System
<https://youtu.be/Qujsd4vkqFI>
2. Explore job information about Business / System Analyst
<http://tinyurl.com/ztzulel>

References

- Baecker, R. M. (2019). *Computers and society: Modern perspectives*: Oxford University Press, USA.
- Kaczmarczyk, L. C. (2016). *Computers and society: computing for good*: CRC Press.
- Stair, R., & Reynolds, G. (2020). *Principles of information systems*: Cengage Learning.
- Litchfield, A. (2017). *INFS500 Enterprise Systems - Bachelor of Computer and Information Science: Process Modelling Workbook*. Auckland: Auckland University of Technology.
- Symons, C. R. (1991). *Software sizing and estimating: Mk II FPA (function point analysis)*: John Wiley & Sons, Inc.