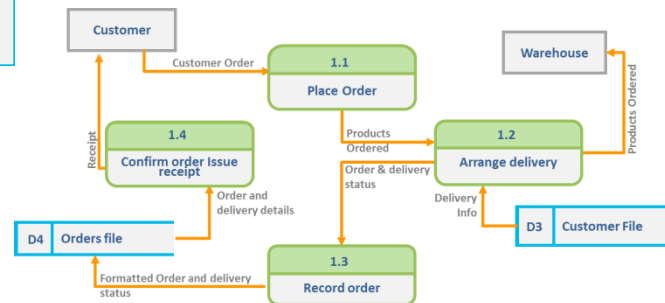
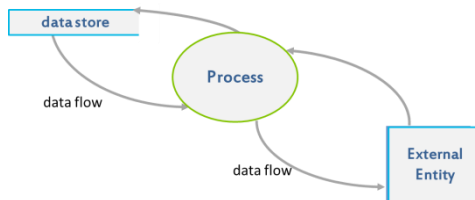


DIS501

Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων

Lecture 10: Δομημένη Ανάλυση και Σχεδιασμός (SSAD)

*Μοντελοποίηση Διαδικασιών και
Διαγράμματα Ροής Δεδομένων
(Process Modelling and Data Flow
Diagrams)*



By: Dr. Avgousta Kyriakidou-Zacharoudiou



Learning Objectives

- Κατανόηση της φιλοσοφίας και των βασικών αρχών της Δομημένης Ανάλυσης και Σχεδίασης Συστημάτων
- Ανάλυση ομοιοτήτων και διαφορών ανάμεσα στη SSAD και την Αντικειμενοστραφή Ανάλυση και Σχεδίαση (OOAD)
- Ερμηνεία της σημασίας των λογικών και φυσικών διαγραμμάτων ροής δεδομένων (DFDs) για την απεικόνιση της ροής πληροφορίας ανάμεσα σε ανθρώπους και συστήματα σε έναν οργανισμό
- Σχεδίαση, χρήση και ανάλυση λογικών DFDs με τεχνικές ιεραρχικής αποδόμησης (levelling) για την κατανόηση του υφιστάμενου συστήματος
- Δημιουργία φυσικών DFDs με βάση τα λογικά DFDs που έχουν ήδη αναπτυχθεί



So far...

- **Soft Systems Methodology**

- Εξετάστηκε η προβληματική κατάσταση (rich picture)
- Καθορίστηκαν (σχετικά) συστήματα προς δημιουργία – Root definitions
- Μοντελοποιήθηκαν και συζητήθηκαν
- Ελήφθη απόφαση για το αν είναι υλοποιήσιμα
- Επιλέχθηκε το σύστημα που θα αναπτυχθεί

- **Καθορισμός Απαιτήσεων (Requirements Determination)**

- Καταγράφηκαν οι απαιτήσεις – λειτουργικές και μη λειτουργικές
- Εξετάστηκαν ζητήματα feasibility και πιθανοί τρόποι υλοποίησης του συστήματος, και επιλέχθηκε ένας
- Παράχθηκε περιγραφή υψηλού επιπέδου της αναμενόμενης λειτουργικότητας του συστήματος



Requirements Specification

Δυο προσεγγίσεις approaches για μοντελοποίηση των λειτουργικών απαιτήσεων και τον σχεδιασμό του συστήματος

Object Oriented Analysis and design (OOAD)

- Use-case Model – Use-case diagrams
- Object model – Class Diagram
- Dynamic Model – Object Sequence and State Chart Diagrams

2. Structured Systems Analysis and design (SSAD) - Traditional

- Process / Functional model – Data Flow Diagram
- Data model – Entity Relationship Diagram
- Behaviour model – Entity life history



Process Modelling

Objectives

- Process Modelling
 - Data Flow Diagrams – DFDs
 - DFD Notation
 - DFD Levelling (Levels of Abstraction), and Balancing
 - Connectivity & Validation - Evaluating DFD Quality



SSAD

- Η αντικειμενοστραφής ανάλυση και σχεδίαση (OOAD) εμφανίστηκε μόλις στα τέλη της δεκαετίας του 1980.
- Από τα τέλη της δεκαετίας του 1970 και μέσα στη δεκαετία του 1990, μια εναλλακτική προσέγγιση στην ανάλυση καθιερώθηκε ως η κυρίαρχη:
 - **η λεγόμενη Δομημένη Ανάλυση και Σχεδίαση Συστημάτων (SSAD).**

Ακόμα αν σήμερα έχουμε υιοθετήσει σε μεγάλο βαθμό την αντικειμενοστραφή προσέγγιση (OOAD), είναι σημαντικό να γνωρίζουμε τουλάχιστον τα βασικά γύρω από τη Δομημένη Ανάλυση και Σχεδίαση Συστημάτων.



SSAD

WHY???

- Εξακολουθεί να αποτελεί τη βάση για τη διαμόρφωση των περισσότερων ανθρώπων στον τομέα της ανάπτυξης συστημάτων.
- Διδάσκεται για πάνω από τρεις δεκαετίες – υπάρχουν πολλοί επαγγελματίες για τους οποίους η δομημένη προσέγγιση είναι συνώνυμη με την ανάλυση συστημάτων.
- Ιδέες της SSAD, όπως τα διαγράμματα ροής δεδομένων, έχουν εξελιχθεί σε κοινή γλώσσα στον τομέα.

«Πιθανότατα δεν υπάρχει άλλη μεθοδολογία μηχανικής λογισμικού που να έχει προκαλέσει τόσο μεγάλο ενδιαφέρον, να έχει εφαρμοστεί (και συχνά απορριφθεί και στη συνέχεια επαναχρησιμοποιηθεί) από τόσους πολλούς ανθρώπους, να έχει προκαλέσει τόση κριτική και να έχει πυροδοτήσει τόση συζήτηση. Παρ' όλα αυτά, άντεξε στον χρόνο και έχει αποκτήσει σημαντική υποστήριξη στην κοινότητα της μηχανικής λογισμικού.» (Pressman)



SSAD

Η Δομημένη Ανάλυση και Σχεδίαση Συστημάτων (SSAD) αποτελεί μία από τις βασικές εξελικτικές γραμμές στα μοντέλα διαδικασιών.

- Βασίζεται σε μια προσέγγιση γραμμικού κύκλου ζωής (linear life cycle),

Στον πυρήνα της, η SSAD ενσωματώνει τη λογική του **μοντέλου «καταρράκτη»**. Θεωρεί ότι οι απαιτήσεις του συστήματος είναι σταθερές και σαφώς καθορισμένες εξ αρχής.

- Δύσκολη την ενσωμάτωση αλλαγών στις προδιαγραφές και συνεπώς περιορίζει τη δυνατότητα για επαναληπτική ανάπτυξη και διαδοχικές βελτιώσεις στα διαγράμματα.
- Απαιτείται περισσότερος έλεγχος και δοκιμές στο τέλος κάθε φάσης.
- Τα διαγράμματα που χρησιμοποιούνται δεν είναι εμπλουτισμένα με λεπτομερή πληροφορία και δεν μετατρέπονται εύκολα σε κώδικα → δυσκολεύει τη συντήρηση του συστήματος.

Αυτά προκύπτουν:

- η προσέγγιση της SSAD δεν είναι καλά προσαρμοσμένη σε επαναληπτικές και σταδιακές (incremental) μεθοδολογίες ανάπτυξης.



SSAD

SSAD (Δομημένη Ανάλυση & Σχεδίαση Συστημάτων)

Βασική Ιδέα	Βασίζεται στη διαίρεση του συστήματος σε διαδικασίες και ροές δεδομένων.
Τρόπος Ανάλυσης	Χωρίζει το σύστημα σε λειτουργίες και διαδικασίες (process-driven).
Δομή του Συστήματος	Το σύστημα περιγράφεται με Δομές Δεδομένων, Διαγράμματα Ροής Δεδομένων (DFDs) και Καταστάσεις .
Κύριες Τεχνικές	Χρήση Διαγραμμάτων Ροής Δεδομένων (DFD), Δομές Απόφασης, και ER Διαγράμματα .
Εστίαση	Εστιάζει περισσότερο στο ΤΙ κάνει το σύστημα (λειτουργικότητα).
Επαναχρησιμοποίηση Κώδικα	Χαμηλή επαναχρησιμοποίηση (ο κώδικας γράφεται ξανά για νέα έργα).
Καταλληλότητα	Καλύτερο για παραδοσιακά, στατικά συστήματα που έχουν σαφώς ορισμένες διαδικασίες.

OOAD (Αντικειμενοστραφής Ανάλυση & Σχεδίαση)

Βασίζεται στη μοντελοποίηση του συστήματος ως σύνολο αντικειμένων που αλληλεπιδρούν.
Χωρίζει το σύστημα σε αντικείμενα με δεδομένα και συμπεριφορές (object-driven).
Το σύστημα περιγράφεται με κλάσεις, αντικείμενα και σχέσεις μεταξύ τους .
Χρήση Διαγραμμάτων UML , όπως Use Case Diagrams, Class Diagrams, Sequence Diagrams .
Εστιάζει στο ΠΩΣ τα αντικείμενα αλληλεπιδρούν και εξελίσσονται .
Υψηλή επαναχρησιμοποίηση μέσω κληρονομικότητας (inheritance) και πολυμορφισμού (polymorphism) .
Ιδανικό για πολύπλοκα, δυναμικά συστήματα όπου η αλληλεπίδραση μεταξύ αντικειμένων είναι σημαντική.



SSAD

Models and Diagrams

- **Context Diagram:** εξωτερική άποψη του νέου συστήματος
- **Data Flow Diagram:** απεικονίζει τις διαδικασίες που εκτελούνται μέσα στο σύστημα
 - Κάθε επιμέρους διαδικασία μπορεί να αναλυθεί περαιτέρω σε μικρότερου επιπέδου προδιαγραφές
 - αποτυπώνεται με χρήση της μορφής των Δομημένων Αγγλικών (Structured English), γνωστής και ως **MiniSpecs**
- **Data Model (e.g. Entity Relationship Diagram):** στατική δομή του συστήματος, δηλαδή πώς οργανώνονται και σχετίζονται μεταξύ τους τα δεδομένα.
- **Entity Life Histories (ELH) or State Transition diagrams :** τεχνικές που βοηθούν να καταλάβουμε την συμπεριφορά των οντοτήτων του συστήματος



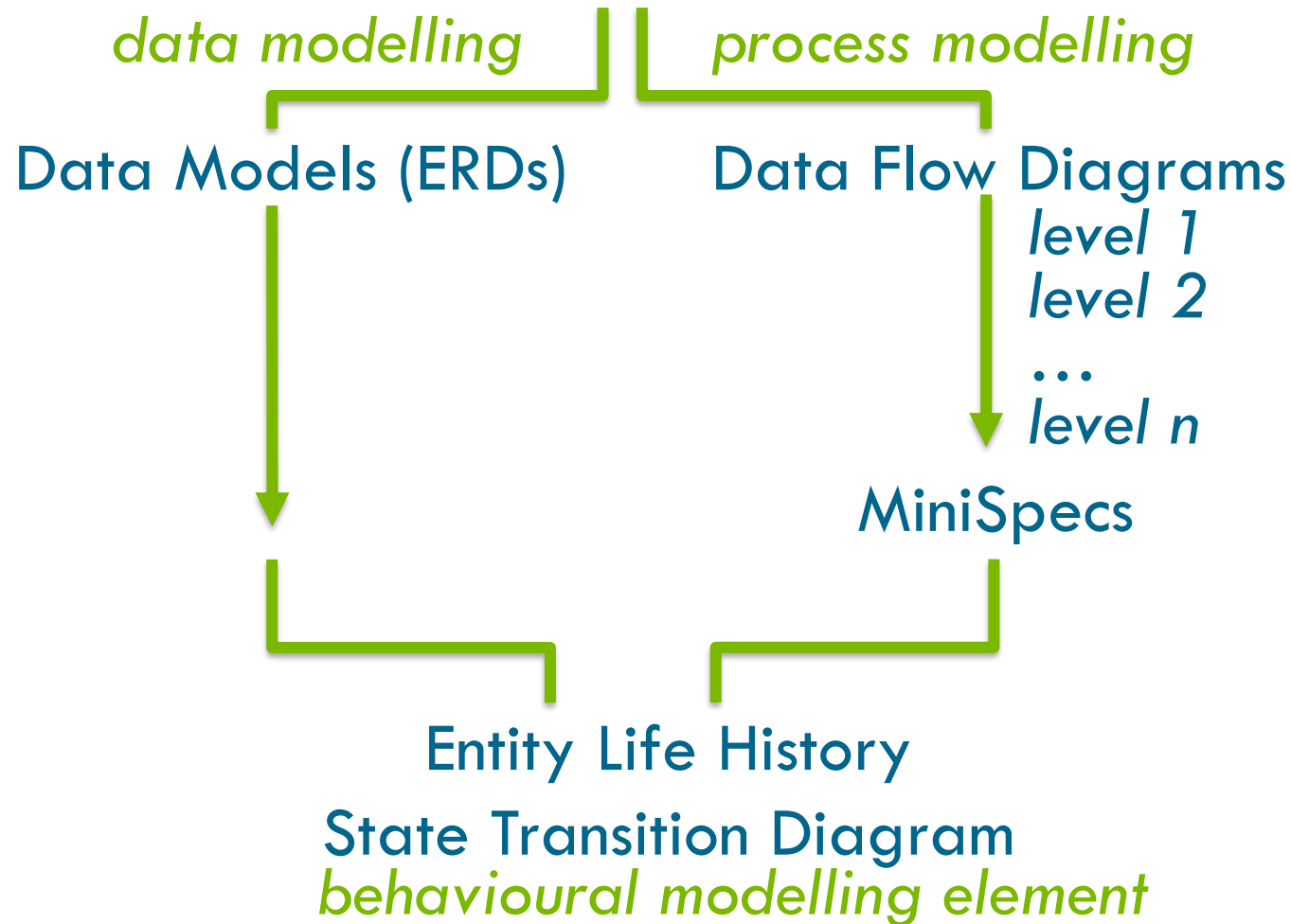
SSAD

Models and Diagrams

- Ένας άλλος τρόπος να δούμε το δομημένο μοντέλο είναι ως ένα **ιεραρχικό σύνολο υπομοντέλων**:
 - ανάλυση του προβλήματος σε πτυχές «δεδομένων» και «διαδικασιών»
 - και τελικά επανασύνδεση (ή διασταύρωση) αυτών των δύο πτυχών στις Ιστορίες Ζωής Οντοτήτων (ELH)



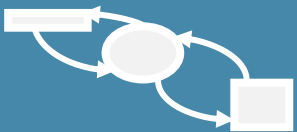
Structured System Modelling





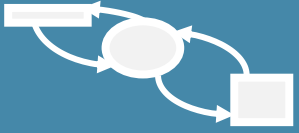
Process Modelling – Creating Data Flow Diagrams





Process/functional modelling

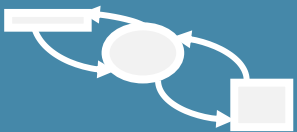
- Όταν αναλύουμε πώς λειτουργεί μια επιχείρηση, συνήθως ξεκινάμε από τις διαδικασίες της — δηλαδή τις δραστηριότητες και ενέργειες που την απαρτίζουν.
 - Μια **διαδικασία** μπορεί να είναι οτιδήποτε προκαλεί αλλαγή ή προσθέτει αξία, όπως η καταχώρηση μιας παραγγελίας ή η εξυπηρέτηση ενός πελάτη
- Η **μοντελοποίηση διαδικασιών** είναι ένας τρόπος να αποτυπώσουμε με ακρίβεια πώς λειτουργεί ένα σύστημα σε μια επιχείρηση
 - τι λειτουργίες υποστηρίζει
 - πώς ρέουν τα δεδομένα ανάμεσα στα επιμέρους μέρη του και πώς οι διαδικασίες σχετίζονται με τα δεδομένα
 - Αποτυπώνονται μέσω Data Flow Diagrams (DFD) τα οποία δείχνουν τη σύνδεση ανάμεσα στις διαδικασίες και στις οντότητες που έχουν καταγραφεί σε διαγράμματα όπως τα ERDs



Process/functional modelling

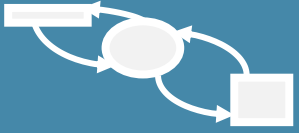
- Ένα σύστημα λειτουργεί λαμβάνοντας **είσοδο** από το περιβάλλον, **μετασχηματίζοντας** (επεξεργαζόμενο) αυτή την είσοδο και παράγοντας μια **έξοδο**
- Μια διαδικασία είναι μια **δραστηριότητα / εργασία / ενέργεια** που εκτελείται πάνω σε εισερχόμενες ροές δεδομένων είτε ως απόκριση σε κάποιες συνθήκες.
- Ένα σύστημα αποτελείται από μια σειρά διαδικασιών





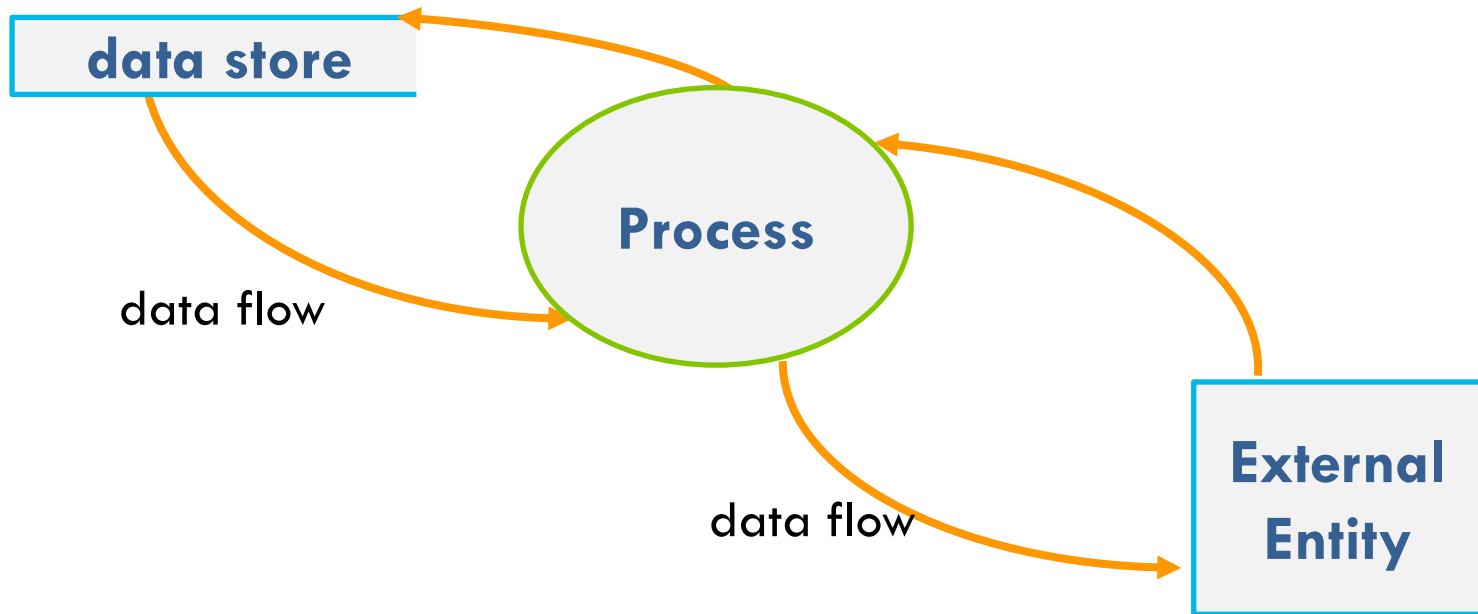
Data Flow Diagrams – DFDs

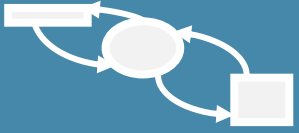
- Καταγράφουν τις λειτουργίες / δραστηριότητες που εκτελεί το σύστημα ως διαδικασίες.
- Περιγράφουν πώς τα δεδομένα διακινούνται και μετασχηματίζονται τόσο μέσα στο σύστημα όσο και στα όριά του, εστιάζοντας σε:
 - τις σχέσεις και την αλληλεπίδραση μεταξύ των επιμέρους διαδικασιών που το αποτελούν
 - τη ροή των δεδομένων, δηλαδή πώς αυτά μεταφέρονται από τη μία διαδικασία στην άλλη
- Στα όρια του συστήματος:
 - εξωτερικές οντότητες, eg. a user, patient, VisaCheck
 - ΔΕΝ είναι το ίδιο με μια οντότητα σε ER διάγραμμα - παρόμοιο με τους UML actors (στο use case diagram)
- Εντός των ορίων του συστήματος:
 - **Processes** - διαδικασίες (που ενεργούν στα δεδομένα, e.g. check customer credit)
 - **Data flows** - ροές δεδομένων (μεταφορά δεδομένων, eg. store record in file)
 - **Data Stores** - αποθήκες δεδομένων (διατηρούν δεδομένα για μελλοντική χρήση, eg. customer file)



DFD - Data Flow Diagrams

- Έμφαση στην **κίνηση** των **δεδομένων (movement of data)** μεταξύ των **εξωτερικών οντοτήτων (external entities)**, των **διαδικασιών (processes)** και μεταξύ των **διαδικασιών** και των **αποθηκών δεδομένων (data stores)**





Data Flow Diagrams – DFDs

notation

DE MARCO & YOURDON

ANE & SARSON

Process



Data Flow

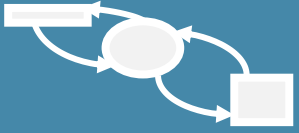


Data Store



External Entity
(source/sink)





Data Flow Diagrams – DFDs

notation - duplicates

DE MARCO & YOURDON

ANE & SARSON

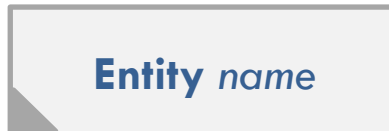
Process

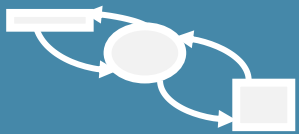


Data Store



External
Entity

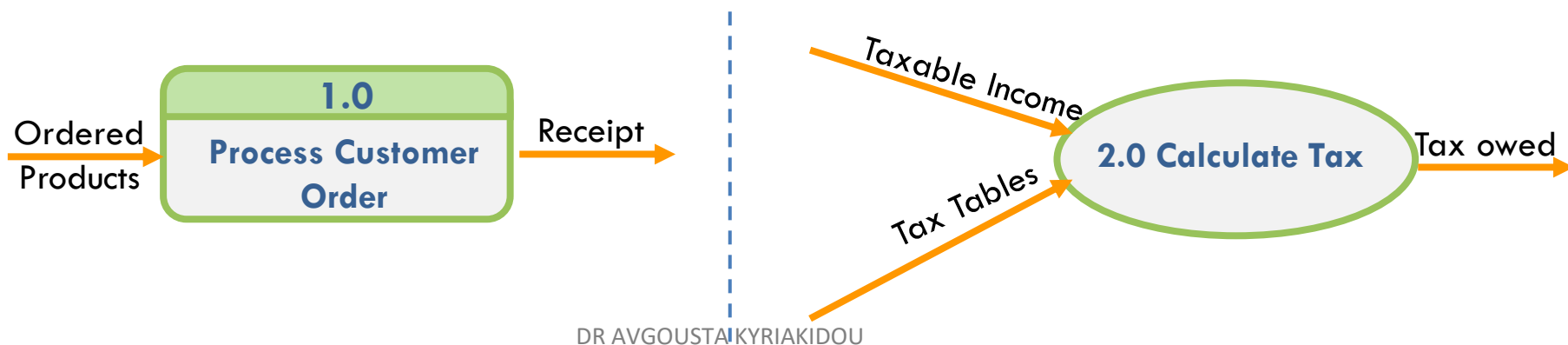


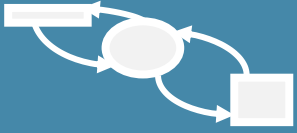


DFD

processes

- Ατομικές δραστηριότητες/λειτουργίες του συστήματος (οι λειτουργικές απαιτήσεις), οι οποίες όταν ολοκληρωθούν με συγκεκριμένη σειρά, επιτυγχάνουν τον συνολικό στόχο του συστήματος
- Μετατρέπουν την είσοδο δεδομένων σε έξοδο**
- Συνδέονται μεταξύ τους μέσω **ροών δεδομένων (data flows)** προς άλλες διαδικασίες, αποθήκες δεδομένων ή εξωτερικές οντότητες

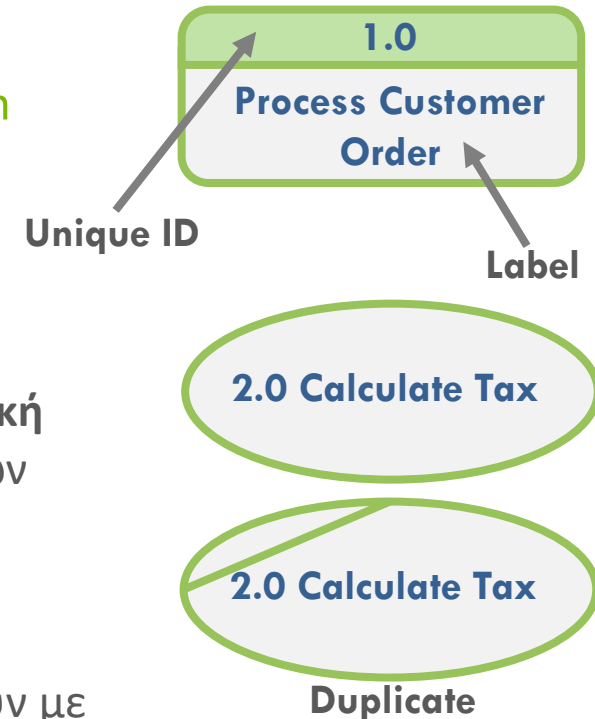


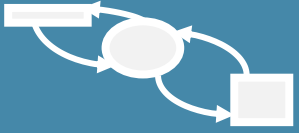


DFD - Notation

processes

- Κάθε διαδικασία έχει ένα **unique identifier and label** (ονομασία)
 - Το όνομα της διαδικασίας πρέπει να είναι Verb Noun combination Labels
 - ο Οι περιγραφές πρέπει να είναι ξεκάθαρες
 - ο eg. register member, process transactions, generate reports, etc.
 - Identifier numbers χρησιμοποιούνται **μόνο για μοναδική αναγνώριση**, μία φορά μόνο στο σύνολο διαγραμμάτων
- Όλες οι διαδικασίες:
 - Πρέπει να **συνδέονται** με κάτι – μπορούν να **συνδεθούν** με άλλες διαδικασίες, αποθήκες δεδομένων ή εξωτερικές οντότητες
 - Πρέπει να έχουν είσοδο και έξοδο – **καμία διαδικασία δεν έχει μόνο εισροές ή μόνο εκροές**

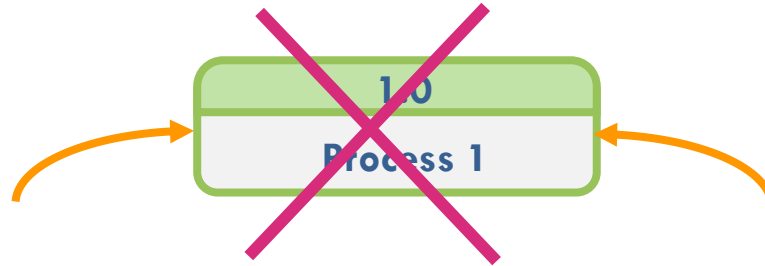




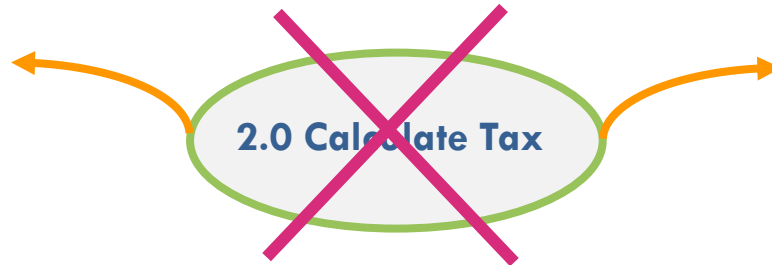
DFD Notation

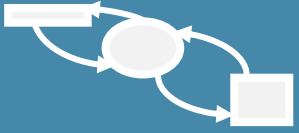
Processes **DON'TS**

- Black Hole



- Miracle





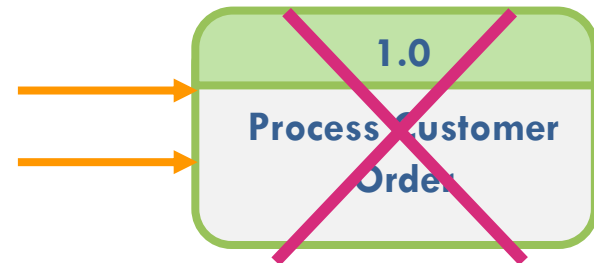
DFD Notation – Process examples

a)



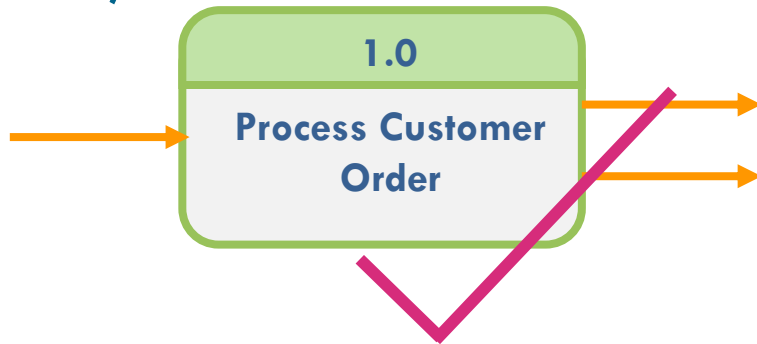
Miracle

b)

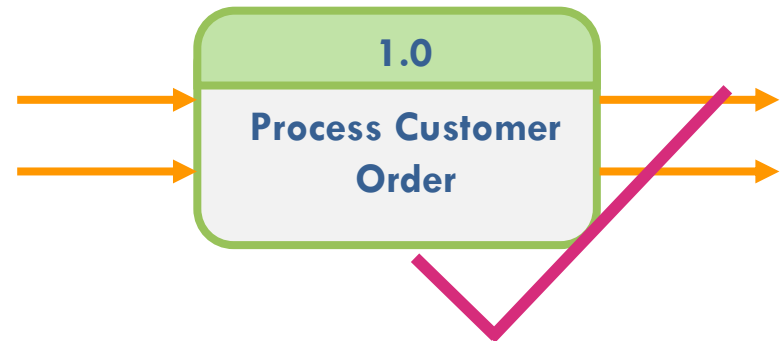


Black Hole

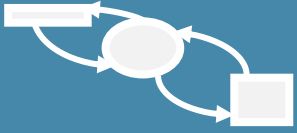
c)



d)



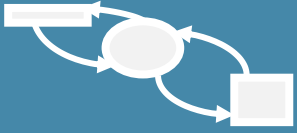
Ποια είναι τα σωστά?



DFD

External Entities

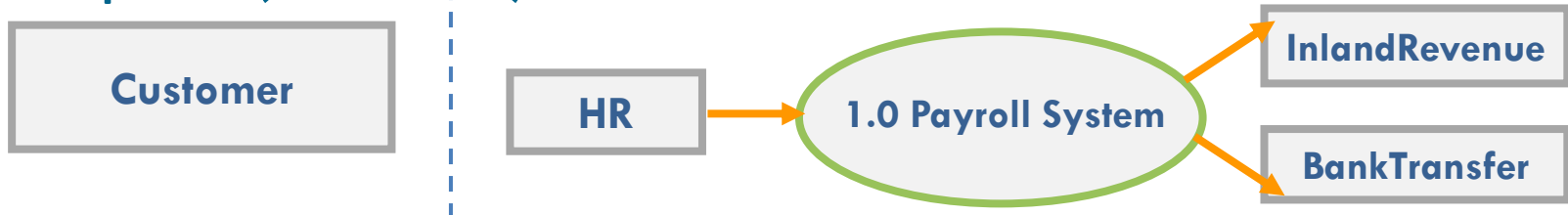
- Οποιοσδήποτε ή οτιδήποτε **εκτός του στόχου του συστήματος**, αλλά που **αλληλεπιδρά με το σύστημα** – κάτι **εξωτερικό ως προς το σύστημα**
- **Λειτουργούν ως πηγές ή προορισμοί (sources or sinks) για τις ροές δεδομένων (dataflows)** που διασχίζουν τα όρια του συστήματος
 - δείχνουν την προέλευση ή/και τον προορισμό των δεδομένων
 - το όνομα υποδεικνύει ποιος είναι ο εξωτερικός πράκτορας
 - επειδή είναι εξωτερικοί, πολλές από τις ιδιότητές τους **δεν μας ενδιαφέρουν**
- Examples of External Entities
 - **Άνθρωποι**: user, patient, admin, passenger, travel agent, supplier, etc.
 - **Οργανισμοί ή τμήματα οργανισμών**: bank, HR department, airline company, etc.
 - **Άλλα συστήματα**: visa check system, warehouse system, finance system, etc.

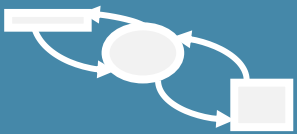


DFD Notation

External Entities

- **Οι εξωτερικές οντότητες πρέπει να ονομάζονται με περιγραφικά, ενικά ουσιαστικά**
 - **Γενικός κανόνας:** Οι εξωτερικές οντότητες τοποθετούνται στα άκρα του DFD, σύμφωνα με τον ορισμό τους ως όρια του συστήματος
 - Οι ροές μεταξύ εξωτερικών οντοτήτων και διεργασιών αναπαριστούν τη διεπαφή ανάμεσα στο σύστημα και τον εξωτερικό κόσμο
 - Οποιαδήποτε σχέση υπάρχει μεταξύ εξωτερικών οντοτήτων δεν πρέπει να φαίνεται στο DFD
- Μπορεί να αναπαραχθούν στο DFD για να αποφευχθεί το μπλέξιμο των γραμμών
- **Οι εξωτερικές οντότητες δεν μπορούν να συνδεθούν άμεσα με αποθήκες δεδομένων (data stores)**

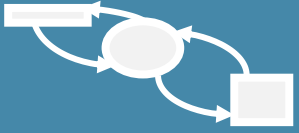




DFD

Data Store

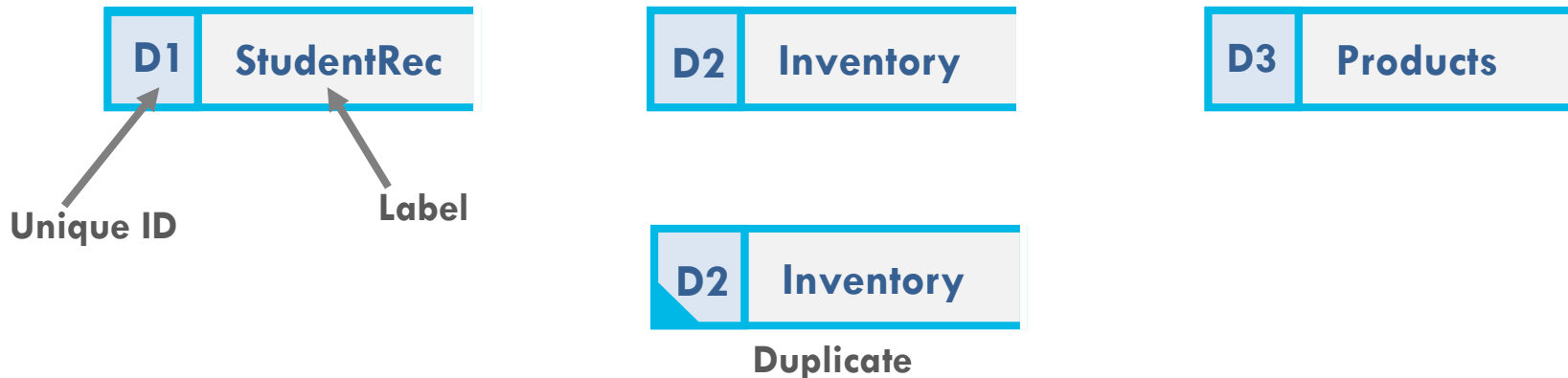
- Ένα προσωρινό ή μόνιμο αποθετήριο δεδομένων για τη διάρκεια του συστήματος ή της διαδικασίας
- Οι αποθήκες δεδομένων πρέπει να περιγράφουν **πράγματα** για τα οποία η επιχείρηση θέλει να αποθηκεύσει δεδομένα
- Οι αποθήκες δεδομένων **είναι παθητικές**: δεν μετασχηματίζουν / δεν επεξεργάζονται δεδομένα
- Οι **διαδικασίες** προσθέτουν ή ανακτούν δεδομένα από αυτές τις αποθήκες
- Μπορούν να αναπαριστούν δεδομένα σε μορφή:
 - αρχείων ή βάσεων δεδομένων υπολογιστή
 - φυσικής αποθήκευσης (χαρτιά, φάκελοι, ντουλάπια κ.λπ.)
- Τα DFD **δεν δείχνουν το φυσικό μέσο αποθήκευσης** (π.χ. δίσκους, χαρτιά) των δεδομένων

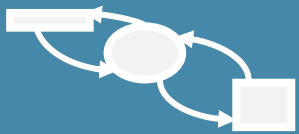


DFD - Notation

Data Store

- Κάθε αποθήκη δεδομένων έχει **unique identifier** and **label**
 - labels πρέπει να είναι ουσιαστικές φράσεις (ουσιαστικά ή περιγραφικοί τίτλοι)
 - Μπορεί να επαναληφθούν στο DFD για να αποφευχθεί η διασταύρωση γραμμών ροής δεδομένων

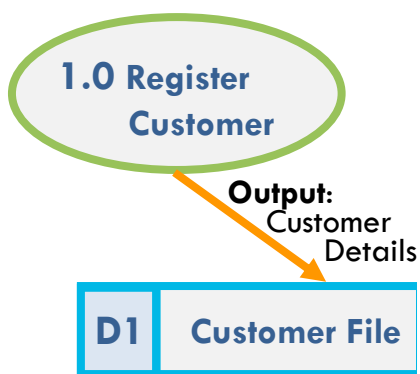




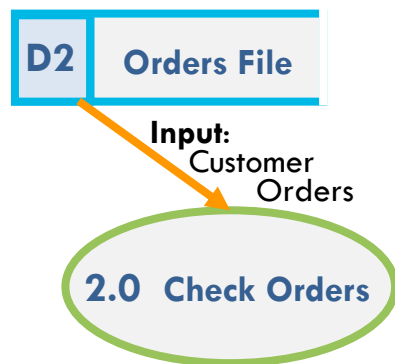
DFD

Data Stores

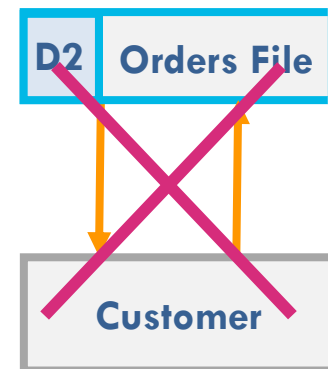
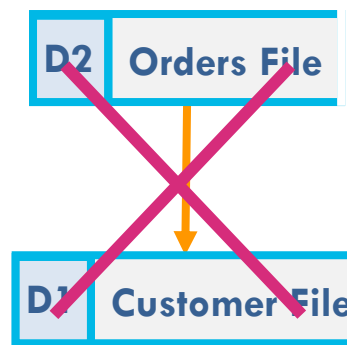
- **Μόνο οι διαδικασίες (processes) συνδέονται με αποθήκες δεδομένων**
 - Η κατεύθυνση της ροής δεδομένων προς ή από μία αποθήκη δείχνει αν τα δεδομένα "διαβάζονται" ή "γράφονται" στη βάση δεδομένων
- Τα **δεδομένα** πρέπει να **μετακινούνται μόνο μέσω κάποιας διαδικασίας**
 - Δεν επιτρέπεται η μεταφορά δεδομένων από μία αποθήκη (data store) σε άλλη χωρίς ενδιάμεση διαδικασία
 - Δεν επιτρέπεται η μεταφορά από εξωτερική οντότητα(external entity) απευθείας σε αποθήκη (data store)

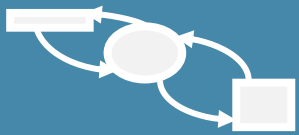


Customer details
stored to data store



Orders read from
data store

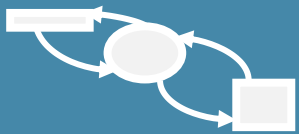




DFD

Data Flow

- Συνδέει διαδικασίες, εξωτερικές οντότητες και αποθήκες δεδομένων (processes, external entities, and data stores)
- **Η ροή δεδομένων (data flow) είναι δεδομένα σε κίνηση**
 - Είναι σαν ένας αγωγός μέσω του οποίου πακέτα πληροφοριών ρέουν από ένα μέρος του συστήματος σε ένα άλλο
 - **Οι ροές δεδομένων είναι οι είσοδοι και έξοδοι των διαδικασιών, των εξωτερικών οντοτήτων και των αποθηκών δεδομένων**
- Οι ροές δεδομένων (data flow) **ΔΕΝ ΕΙΝΑΙ** ροές ελέγχου (π.χ. if-else)

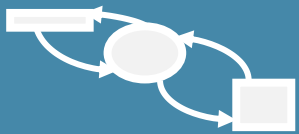


DFD - Notation

Data Flow

- Αναπαρίσταται ως **κατευθυνόμενη γραμμή που απεικονίζει είσοδο ή έξοδο** προς μια διαδικασία, οντότητα ή αποθήκη δεδομένων
- **Πρέπει να έχει ονομασία**
 - Οι ονομασίες πρέπει να είναι περιγραφικά ουσιαστικά ή φράσεις ουσιαστικών
 - Να καθορίζεται η φύση των δεδομένων που κινούνται μέσω της ροής
 - Μπορούν να χρησιμοποιούνται επίθετα και επιρρήματα για να περιγράψουν πώς μεταβλήθηκαν τα δεδομένα κατά την επεξεργασία
 - Καμία ροή δεδομένων δεν πρέπει να μένει χωρίς ετικέτα

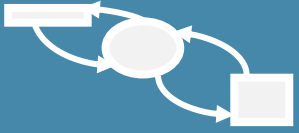




DFD - Notation

Data Flow

- **Όλες οι ροές δεδομένων πρέπει να ξεκινούν ή να καταλήγουν σε μια διαδικασία,** δηλαδή οι ροές δεδομένων συνδέουν ή δείχνουν κατεύθυνση μεταξύ:
 - μιας εξωτερικής οντότητας και μιας διαδικασίας (ή αντίστροφα)
 - μιας διαδικασίας και μιας άλλης διαδικασίας
 - μιας διαδικασίας και μιας αποθήκης δεδομένων (ή αντίστροφα)
- **Αποτελούν είσοδο/έξοδο από αποθήκες δεδομένων**
 - Ροή προς αποθήκη δεδομένων σημαίνει καταχώρηση/ενημέρωση
 - Ροή από αποθήκη δεδομένων σημαίνει ανάκτηση
- **Οι ροές δεδομένων ΔΕΝ συνδέουν ούτε δείχνουν κατεύθυνση μεταξύ:**
 - εξωτερικής οντότητας και άλλης εξωτερικής οντότητας
 - εξωτερικής οντότητας και αποθήκης δεδομένων
 - αποθήκης δεδομένων και άλλης αποθήκης δεδομένων



DFD Examples

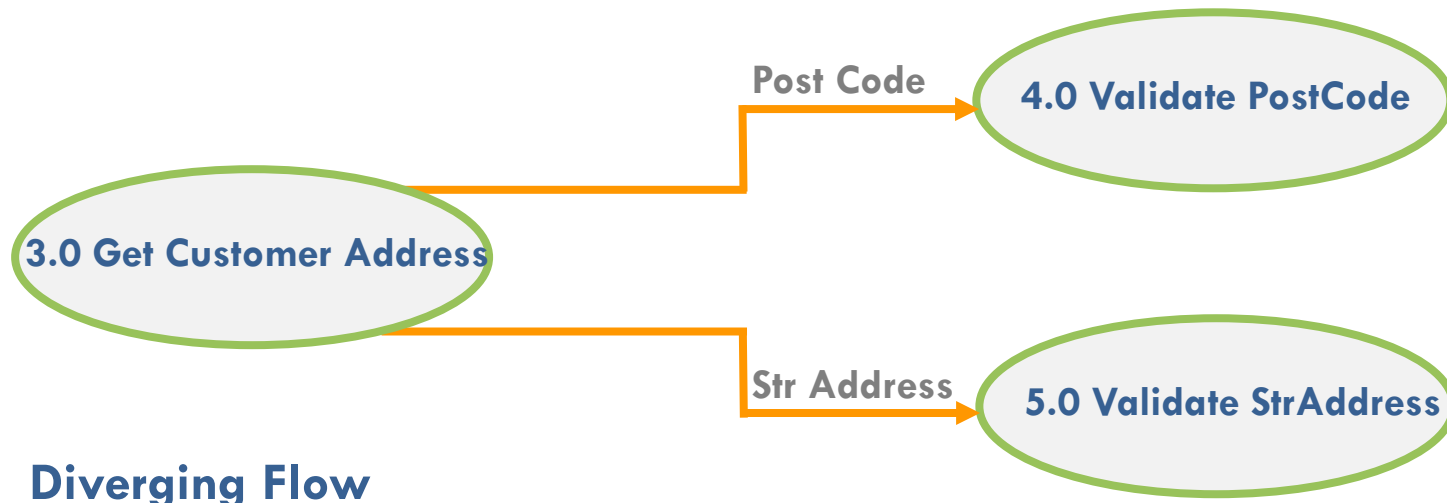
Data Flow



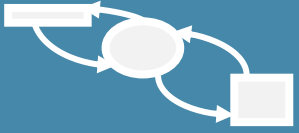
Input Flow



Output Flow



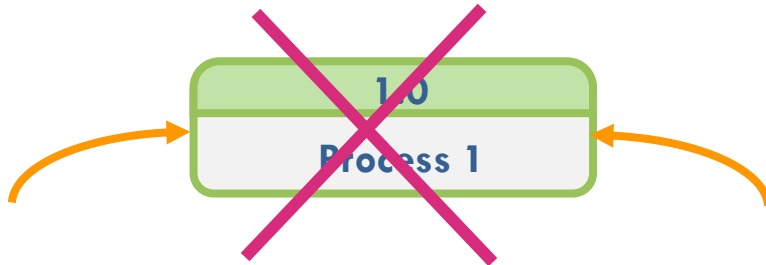
Diverging Flow



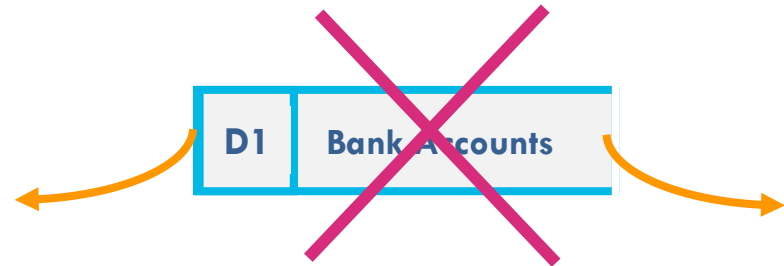
DFD Notation

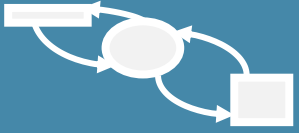
Data Flow **DON'TS**

- Black Hole



- Miracle

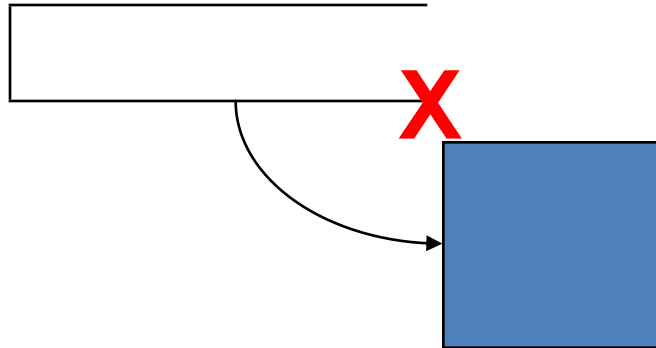




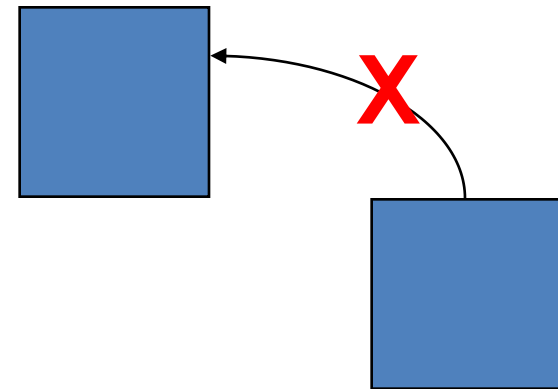
DFD Notation

Data Flow **DON'TS**

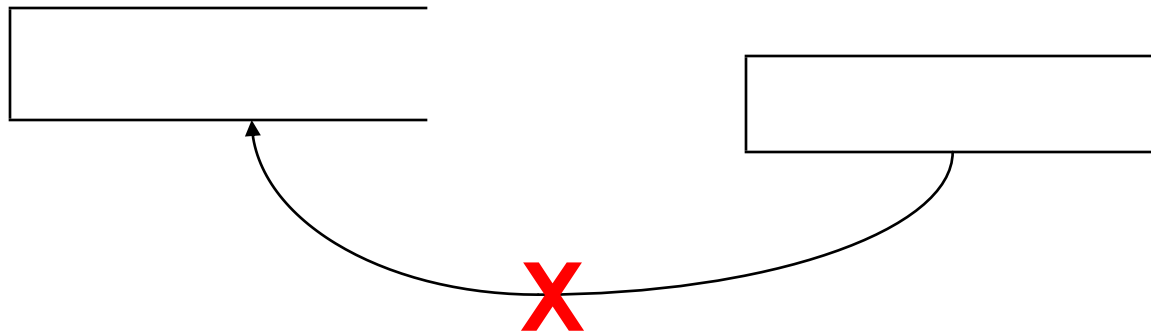
Data store to
Entity - or
reverse

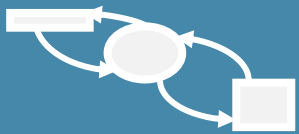


Entity to Entity



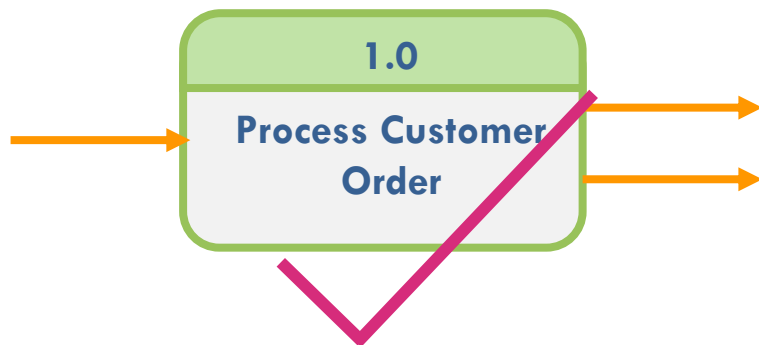
Data store to
Data store



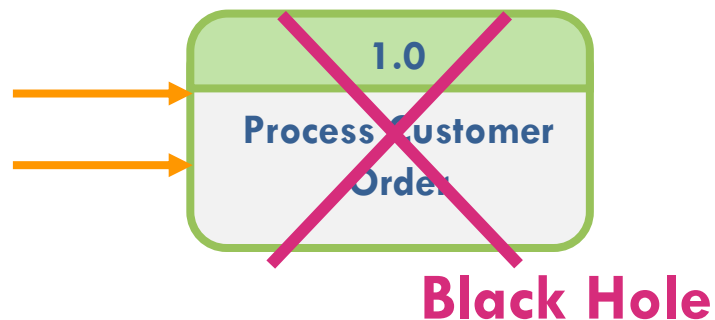


DFD Notation – Process examples

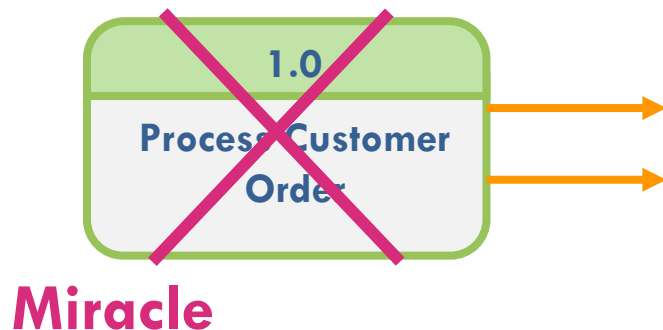
a)



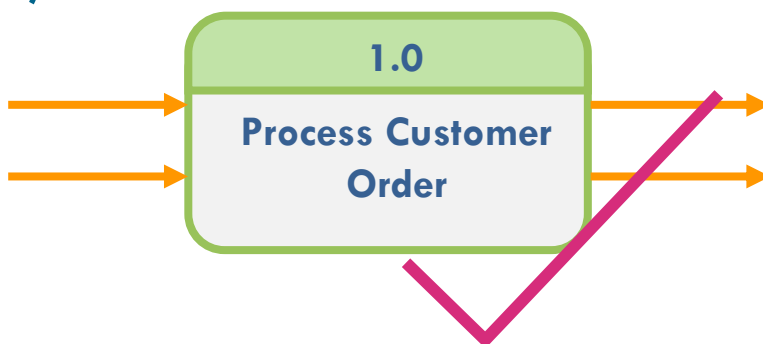
b)



c)



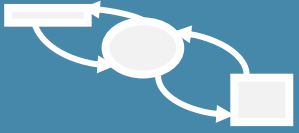
d)



Ποια είναι τα σωστά?



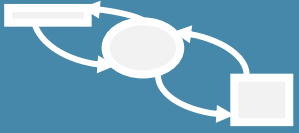
BREAKOUT SESSION



DFD Examples

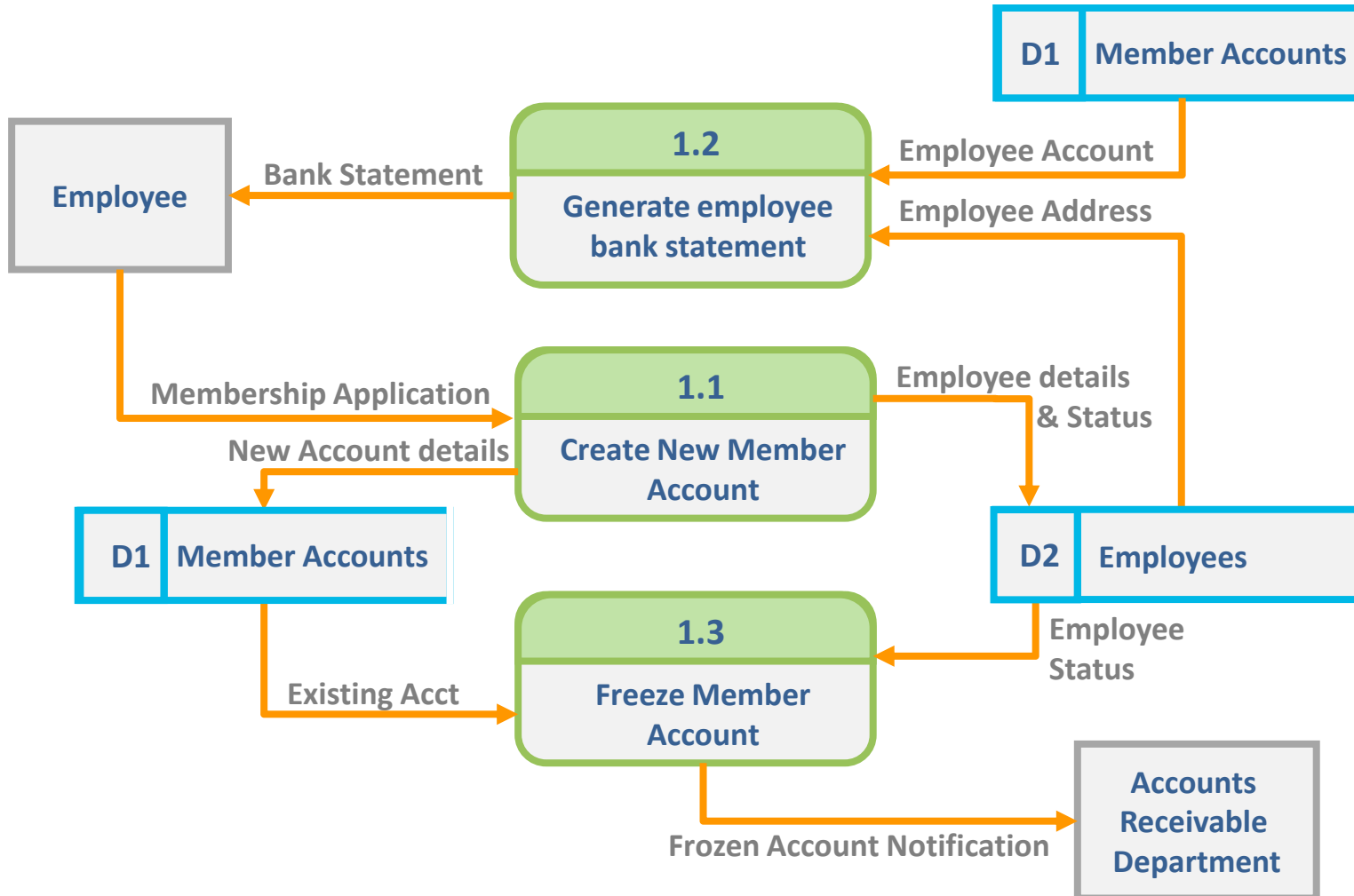
Data Flow

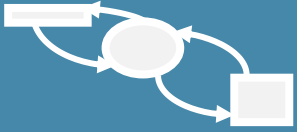




DFD Examples

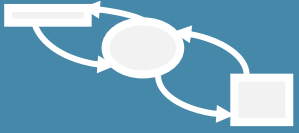
Employee Accounts





DFD Levelling

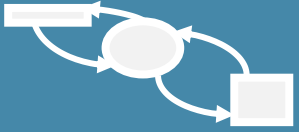
- Τα DFDs επιτρέπουν στον αναλυτή να μοντελοποιήσει και να εξετάσει το σύστημα σε διαφορετικά επίπεδα λεπτομέρειας.
- Μια επιχειρησιακή λειτουργία μπορεί να περιλαμβάνει πολλές διαδικασίες.
- Η απεικόνιση όλων των διαδικασιών σε ένα μόνο διάγραμμα μπορεί:
 - να το κάνει να φαίνεται υπερφορτωμένο,
 - να δυσκολεύει την κατανόηση του τι ακριβώς κάνει κάθε διαδικασία.
- Για να ξεπεραστεί αυτό, είναι σύνηθες να «σπάμε» το διάγραμμα σε επιμέρους επίπεδα, μια διαδικασία γνωστή ως **ιεράρχηση (levelling)**.



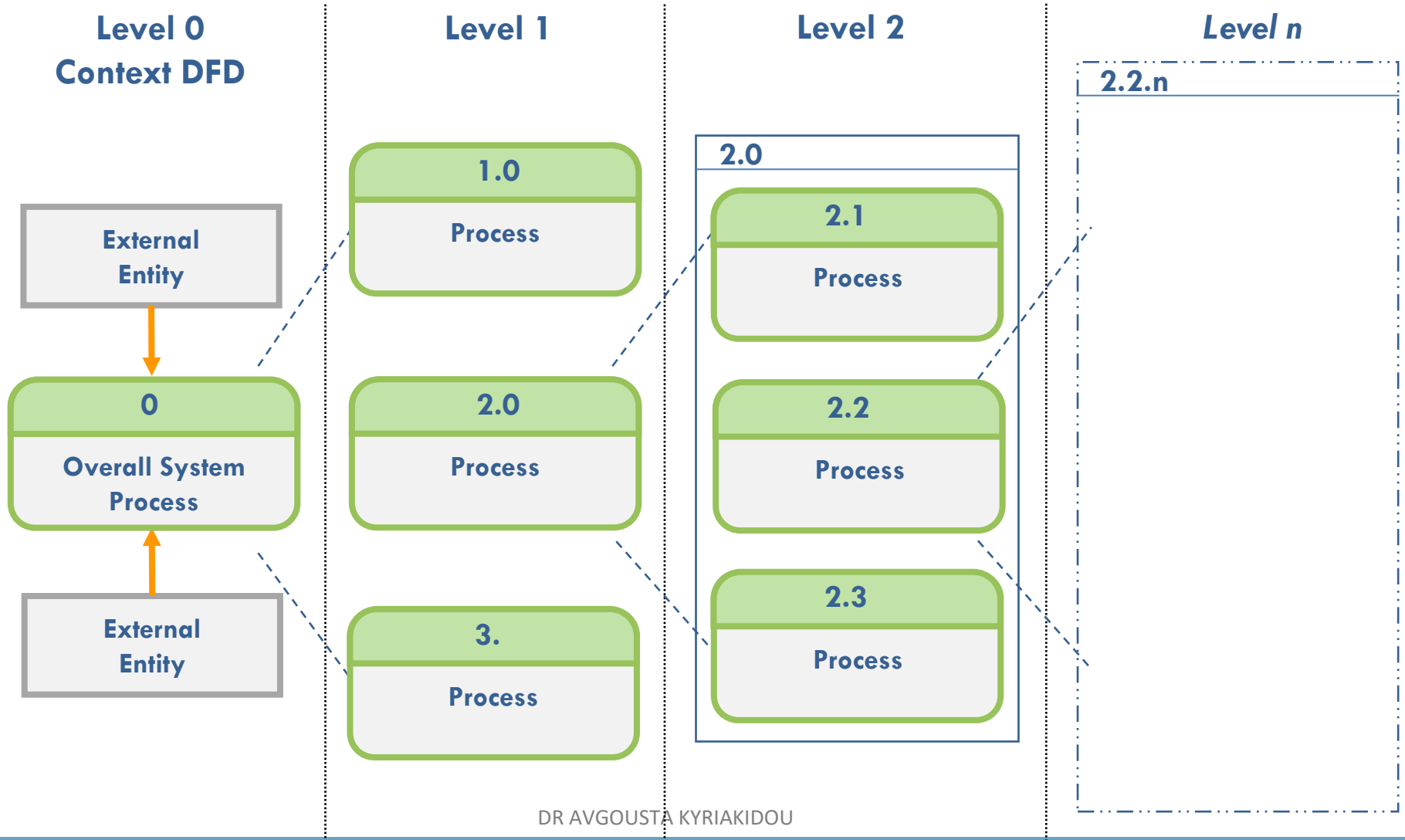
DFD Levelling

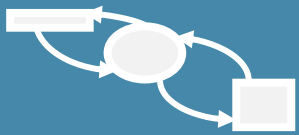
Constructing DFDs

- **Οργάνωση των DFDs σε μια σειρά από επίπεδα**
 - κάθε επίπεδο παρέχει σταδιακά περισσότερες λεπτομέρειες για ένα μέρος του επιπέδου από πάνω
- **Level 0 – Διάγραμμα Πλαισίου (Context Diagram)**
- **Level 1** – Μόλις δημιουργηθεί το Διάγραμμα Επιπέδου 0, παρουσιάζονται περισσότερες λεπτομέρειες για το σύστημα στο Επίπεδο 1
- **Level 2** – Κάθε διαδικασία από το Επίπεδο 1 μπορεί να αναλυθεί σε μικρότερες διαδικασίες και τις ροές δεδομένων μεταξύ τους για παρουσίαση μεγαλύτερης λεπτομέρειας στο Επίπεδο 2
- **Η αποσύνθεση συνεχίζεται μέχρι το σύστημα να περιγράφεται με επαρκή λεπτομέρεια**
- Αυτή η διαδικασία ιεράρχησης των DFDs ονομάζεται εξειδίκευση (**refinement**)
- **Οι διαδικασίες του κατώτατου επιπέδου** (αυτές που δεν μπορούν να αναλυθούν περαιτέρω) ονομάζονται **Πρωτογενείς Διαδικασίες (Primitive Processes)**



DFD Levelling





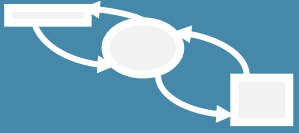
Level 0 DFD

Context Diagram

- Αναγνωρίζει το πεδίο εφαρμογής ενός Πληροφοριακού συστήματος.
- Συνοψίζει όλες τις δραστηριότητες επεξεργασίας για το σύστημα ή υποσύστημα
 - **Υψηλότερο επίπεδο** (πιο αφηρημένη άποψη) του συστήματος

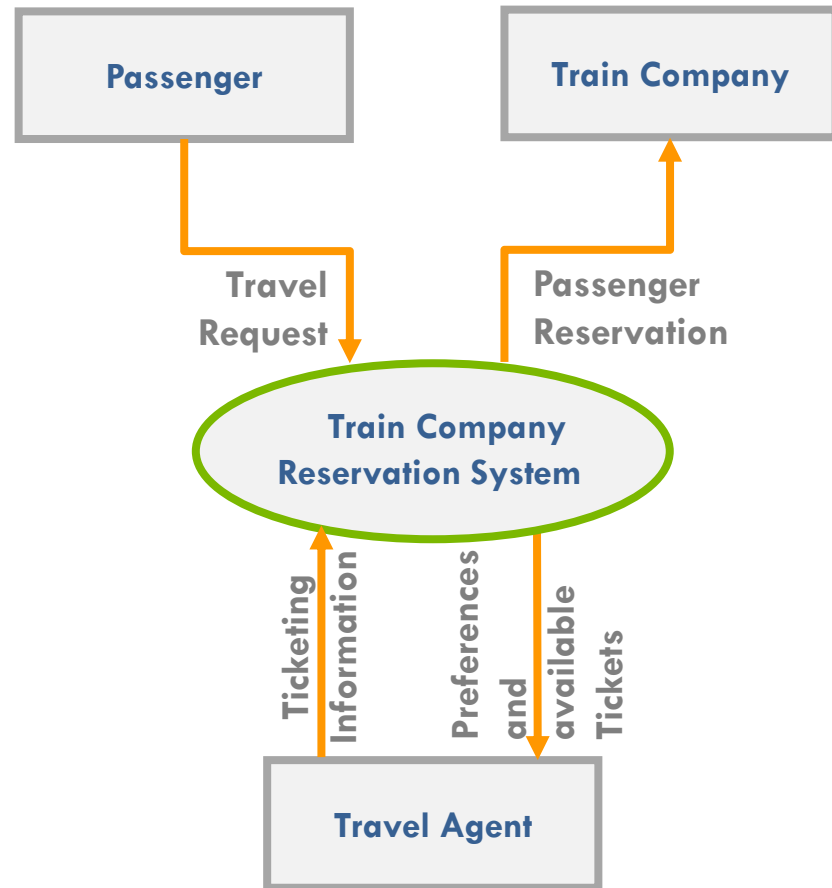
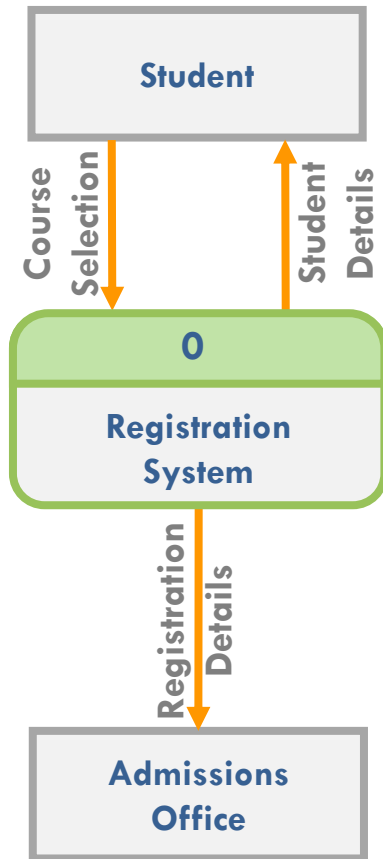
Το Διάγραμμα Πλαισίου δείχνει ολόκληρο το σύστημα ως μία ενιαία διαδικασία

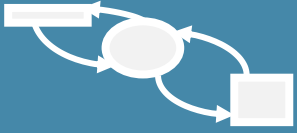
- Περιλαμβάνει:
 - μία διαδικασία (που αναπαριστά ολόκληρο το σύστημα)
 - όλες τις πηγές/αποδέκτες – sources/sinks (εξωτερικές οντότητες)
 - ροές δεδομένων που συνδέουν τη διαδικασία με τις εξωτερικές οντότητες



Level 0 DFD

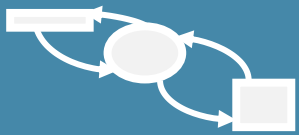
Context Diagram - Examples





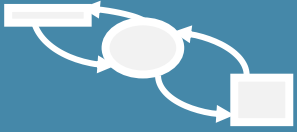
Level 1 DFD

- Ένα διάγραμμα ροής δεδομένων (DFD) που αναπαριστά τις βασικές διαδικασίες, τις ροές δεδομένων και τις αποθήκες δεδομένων ενός συστήματος με υψηλό επίπεδο λεπτομέρειας.
 - Παρουσιάζει όλες τις διαδικασίες που συνθέτουν το συνολικό σύστημα.
 - Δείχνει πώς μετακινείται η πληροφορία από και προς κάθε διαδικασία.
 - Εισάγει τις αποθήκες δεδομένων.



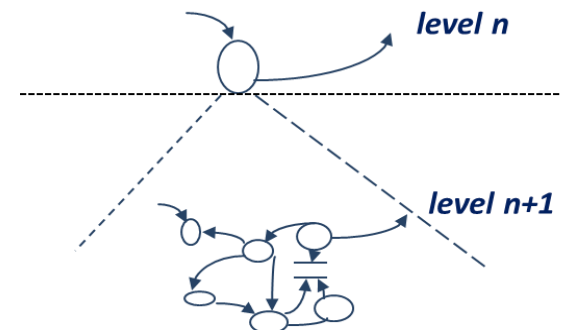
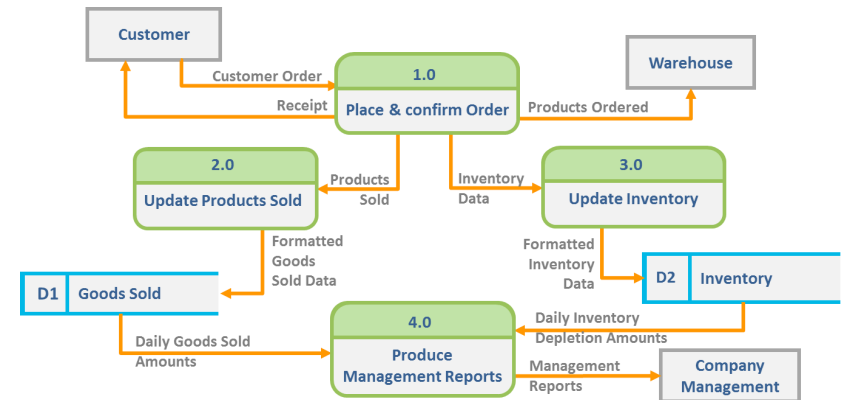
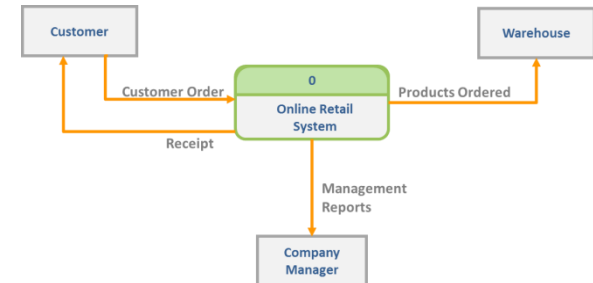
Level 2 DFD

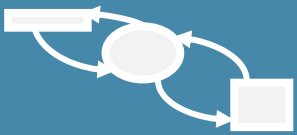
- Παρουσιάζει **όλες τις διαδικασίες** που συνθέτουν **μία μόνο διαδικασία** από το διάγραμμα επιπέδου 1.
- Δείχνει πώς η πληροφορία μετακινείται από / προς καθεμία από αυτές τις διαδικασίες.
- Τα διαγράμματα επιπέδου 2 ενδέχεται να μην είναι απαραίτητα για όλες τις διαδικασίες επιπέδου 1.
- Η σωστή αρίθμηση κάθε διαδικασίας βοηθά στην κατανόηση της θέσης της διαδικασίας στο συνολικό σύστημα.



DFD Levelling

- Πρώτο επίπεδο, μία μόνο διαδικασία – το διάγραμμα περιβάλλοντος (context diagram)
- Περισσότερες λεπτομέρειες προστίθενται στο επίπεδο 1
- Κάθε διαδικασία μπορεί να αναλυθεί περαιτέρω (explode)

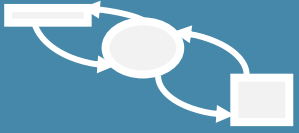




DFD Levelling

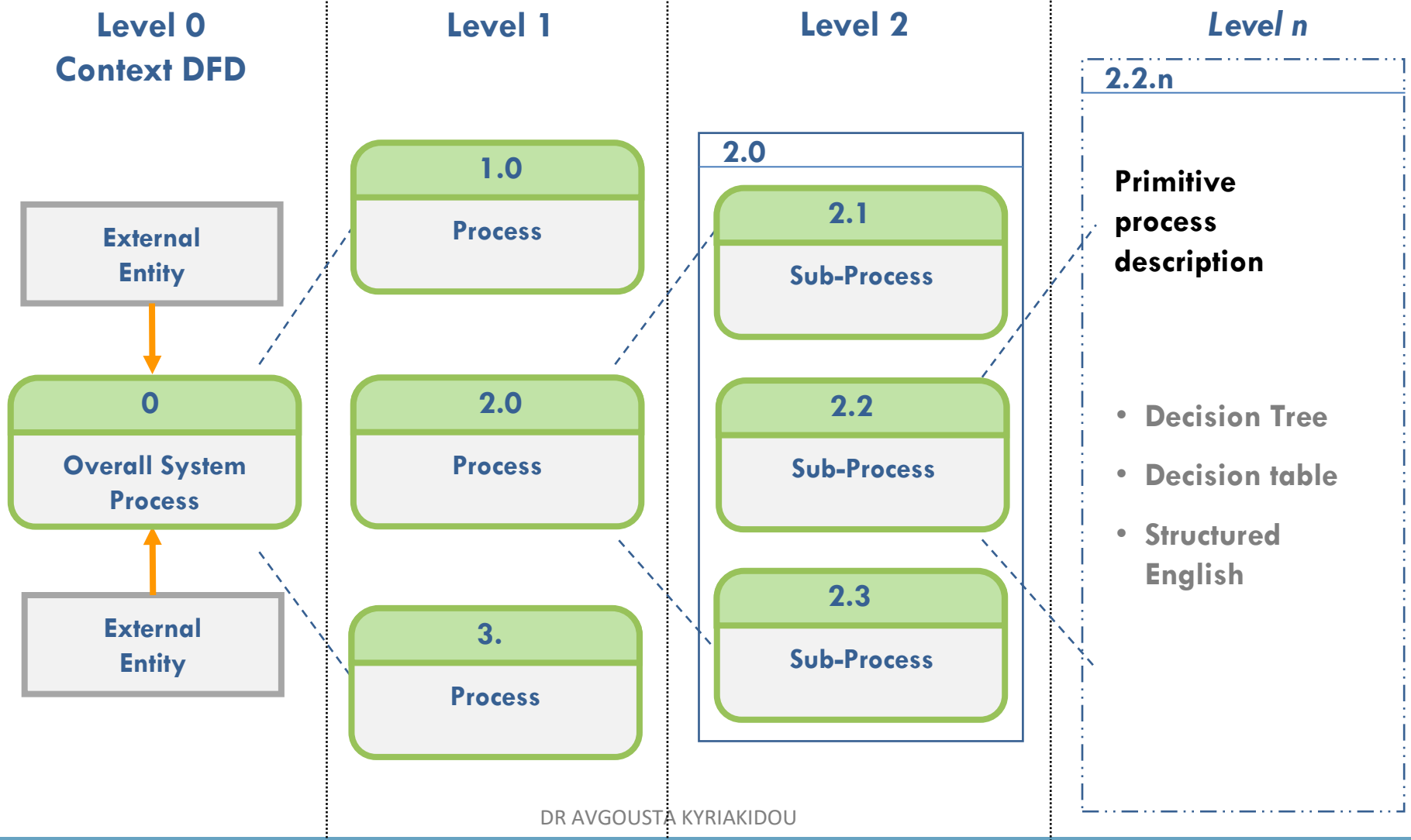
Terminating the levelling process

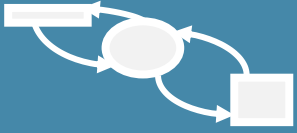
- Η ιεράρχηση (levelling) πρέπει κάποια στιγμή να σταματήσει — οι διεργασίες στο κατώτερο επίπεδο ονομάζονται **πρωτογενείς (primitive processes)** ή **ατομικές (atomic)**.
 - Αυτές οι διεργασίες δεν μπορούν πλέον να διασπαστούν σε μικρότερες.
 - πρέπει να περιγραφούν με διαδικαστικό τρόπο.
- Για να γίνει αυτό, χρησιμοποιούνται τεχνικές όπως οι **minispecs** ή **πίνακες απόφασης**.
- **Minispecs**: Μέσα σε κάθε bottom-level διεργασία του κατώτερου επιπέδου σε ένα DFD, ορίζεται η λειτουργία με λεπτομέρεια.
 - Συνήθως χρησιμοποιείται δομημένη αγγλική γλώσσα.
 - Απλές προτάσεις: περιγράφουν ενέργειες που πρέπει να γίνουν.
 - Σύνθετες προτάσεις: ακολουθούν δομές ελέγχου (if-then-else, loops κ.λπ.).



DFD Levelling

terminal level



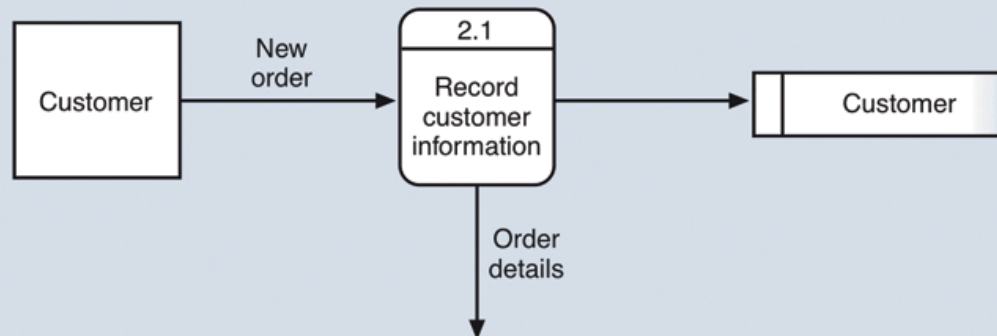


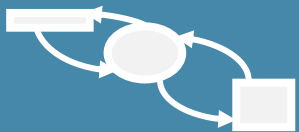
DFD Levelling – Terminal Level

Structured English Process Description

Process 2.1 – Record Customer Information

```
Ask if customer has an account (or has made a previous order)
If customer has an account then
    Ask for identification information
    Query database with identifying information
    Copy query response data to Order details
Else
    Create an empty Customer record in the database
    Ask customer for Customer attributes
    Update empty Customer record with Customer attributes
Endif
Ask customer for order information for first item
While more order items Do
    Update Order details with order information
Endwhile
```





DFD Levelling – Terminal Level

Structured English Process Description vs Use case Scenarios

Use Case Scenario

Minispec

Εστίαση

Εστιάζει στην **αλληλεπίδραση Χρήστη - Συστήματος** (π.χ. τι κάνει ο χρήστης και τι απαντά το σύστημα)

Εστιάζει στο **εσωτερικό της διαδικασίας** — τι βήματα εκτελεί το σύστημα για να υλοποιήσει μια λειτουργία

Οπτική

Οπτική **χρήστη** και του συστήματος μαζί

Οπτική **του συστήματος μόνο**

Τύπος ροής

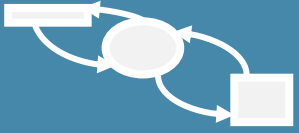
Περιγράφει βήμα-βήμα την **εξωτερική λειτουργική αλληλουχία**

Περιγράφει τη **λογική επεξεργασία δεδομένων** μέσα στη διαδικασία

Σύνδεση με διάγραμμα

Συνδέεται με ένα **Use Case** (στο Use Case Diagram)

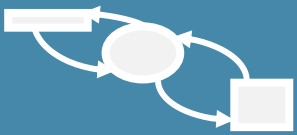
Συνδέεται με ένα **Primitive Process** (στο DFD)



DFDs

Connectivity & Validation

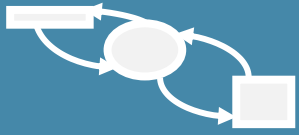
- **Βεβαιωθείτε ότι το Διάγραμμα Επιπέδου 1 είναι πλήρως συνδεδεμένο.**
 - Δηλαδή, μια διαδικασία πρέπει να λαμβάνει εισροές από άλλες διαδικασίες ή από αποθήκες δεδομένων για να ενεργοποιηθεί.
- **Ελέγξτε το Διάγραμμα Επιπέδου 1 ως προς το Context Diagram για συνέπεια:**
 - Οι ροές δεδομένων που διασχίζουν τα όρια πρέπει να είναι ίδιες.
 - Τα ονόματα πρέπει να είναι τα ίδια.



DFDs

tips for constructing

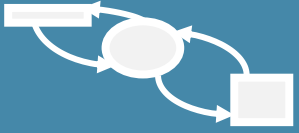
1. **Εντοπίζουμε και καταγράφουμε** όλες τις εξωτερικές οντότητες που αλληλεπιδρούν με το σύστημα, είτε παρέχοντας δεδομένα εισόδου είτε λαμβάνοντας εξόδους.
2. **Καταγράφουμε** τις **ροές δεδομένων** που κατευθύνονται προς ή προέρχονται από **κάθε εξωτερική οντότητα**.
3. **Δημιουργούμε** το διάγραμμα περιβάλλοντος (**Context Diagram**), τοποθετώντας το σύστημα στο κέντρο και συνδέοντας τις εξωτερικές οντότητες μέσω των σχετικών ρών δεδομένων.
4. **Αναγνωρίζουμε** τις **βασικές λειτουργίες (διαδικασίες)** που εκτελούνται εντός του συστήματος.
5. **Καθορίζουμε** τις **ροές δεδομένων** που συνδέουν τις διάφορες **διαδικασίες μεταξύ τους**.
6. **Ανιχνεύουμε και καταγράφουμε** τη διαδρομή κάθε **ροής δεδομένων** που εισέρχεται στο σύστημα — δηλαδή **πού κινείται, πού αποθηκεύεται και πώς μετασχηματίζεται ή επεξεργάζεται**.
7. **Ελέγχουμε** ότι **κάθε ροή δεδομένων έχει σαφώς ορισμένη πηγή και προορισμό**.
8. **Βεβαιωνόμαστε** ότι **καμία ροή δεδομένων δεν εξέρχεται απευθείας από μία αποθήκη χωρίς να περνάει από διαδικασία** — όλα τα δεδομένα πρέπει να διακινούνται μέσω διαδικασιών.



Example: Online Retail System

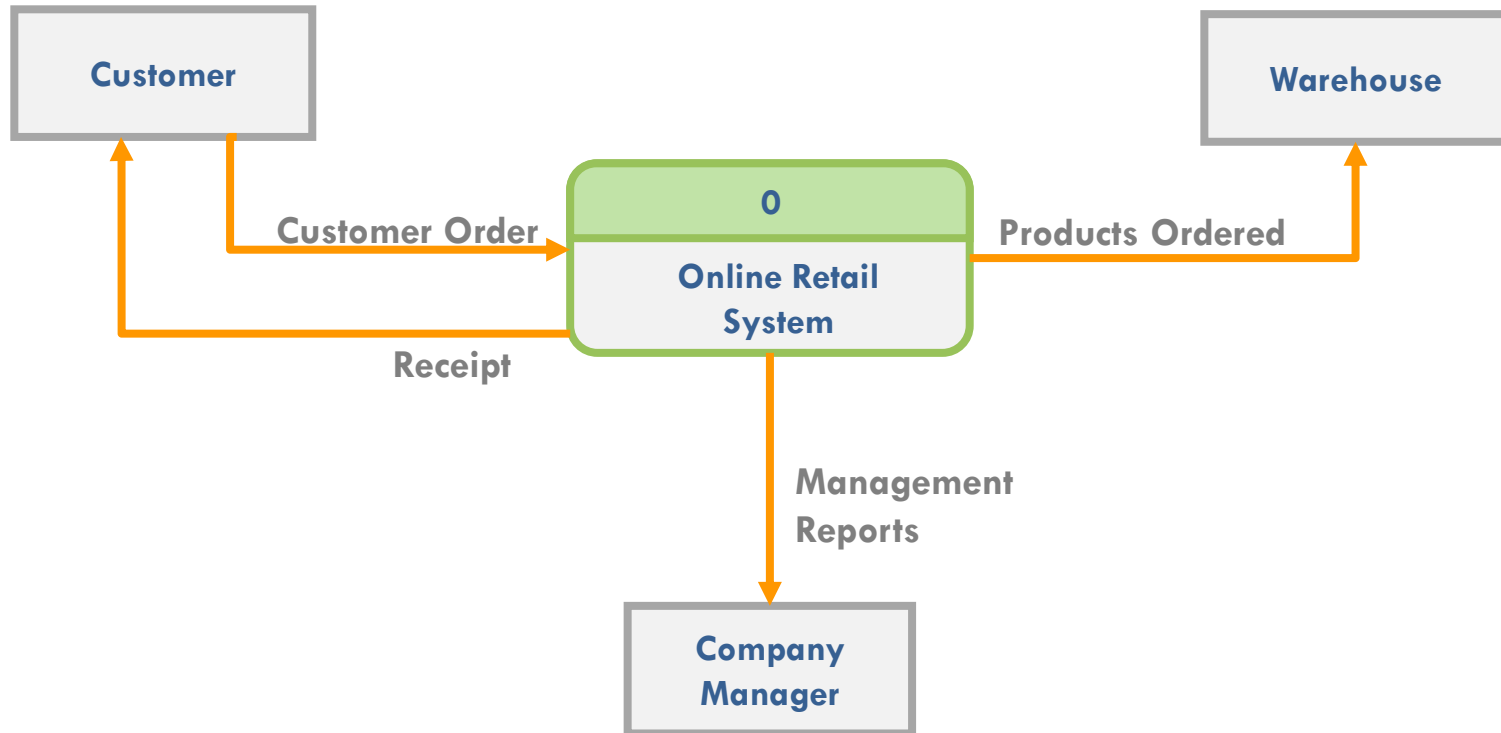
Constructing and Leveling DFDs

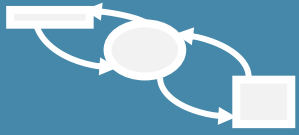
- **Online retail system (functionality subset)**
 - Place orders
 - Process orders
 - Inform warehouse
 - Maintain information about all products sold
 - Maintain inventory
 - Generate management sales reports
- **First Context Diagram - contains no data stores**
 - What do online retail websites do - think of Amazon
 - What are the boundaries
 - Who does it interact with



Example 1: Online Retail System

Context Diagram

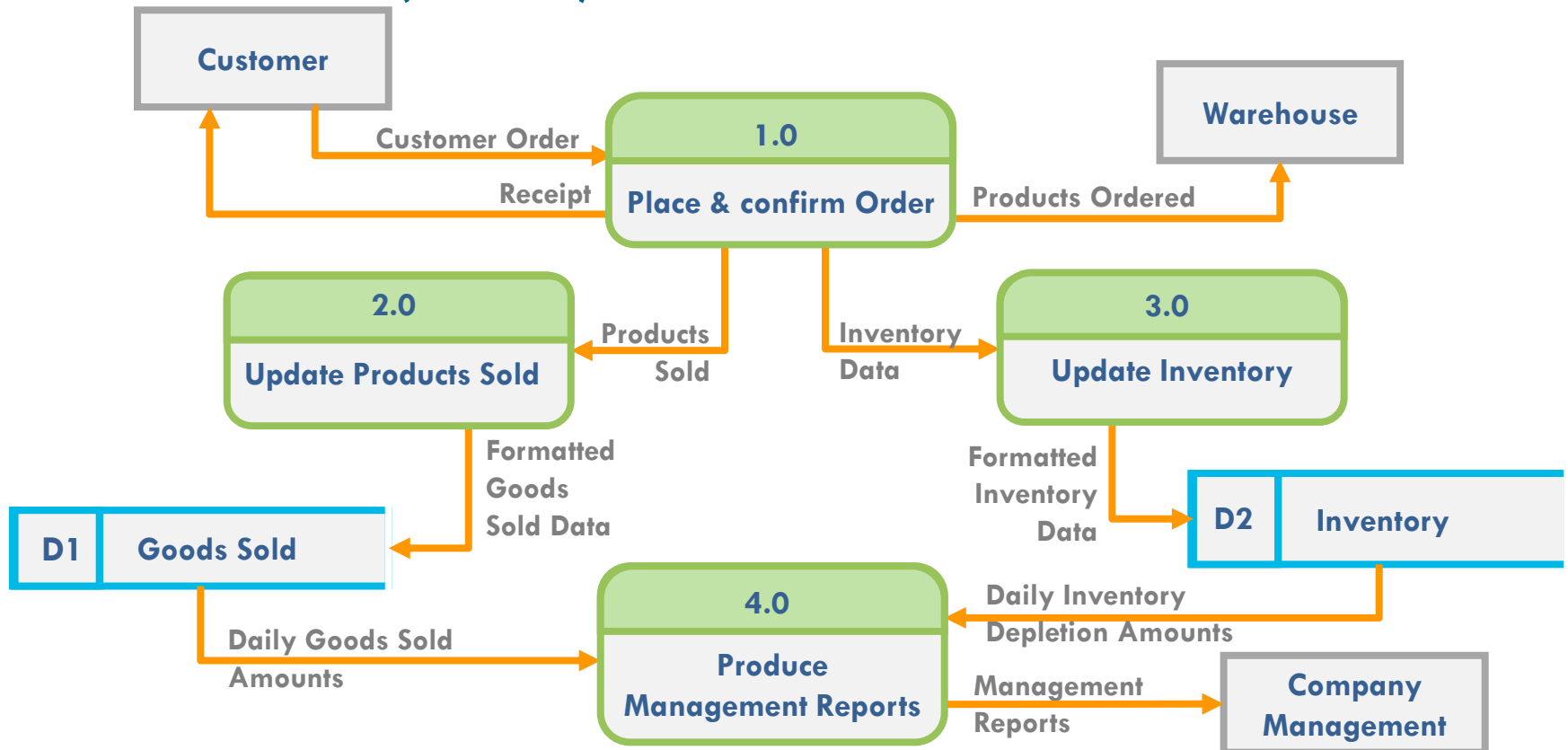


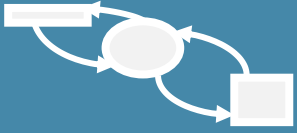


Example 1: Online Retail System

Level 1 DFDs

- Next step is to expand the context diagram to show process breakdown (Level 1)

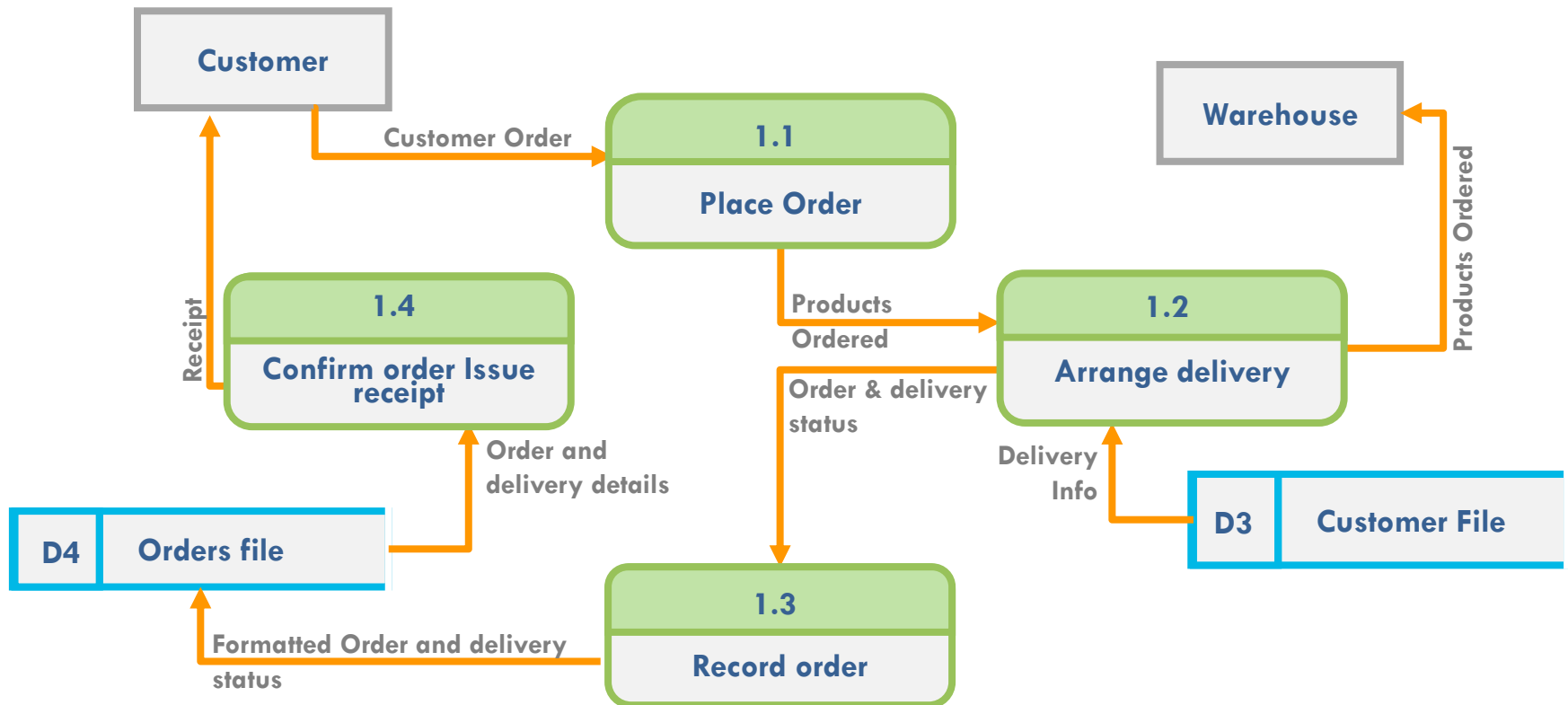


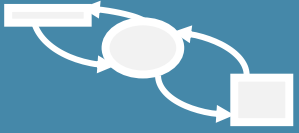


Example 1: Online Retail System

Level 2 DFDs

- Η παραπάνω εικόνα παρουσιάζει ένα Level 2 Διάγραμμα Ροής Δεδομένων (DFD) για το σύστημα Online Retail, και συγκεκριμένα για την ανάλυση της διαδικασίας **1.0 Place & Confirm Order** από το προηγούμενο διάγραμμα Επιπέδου 1.

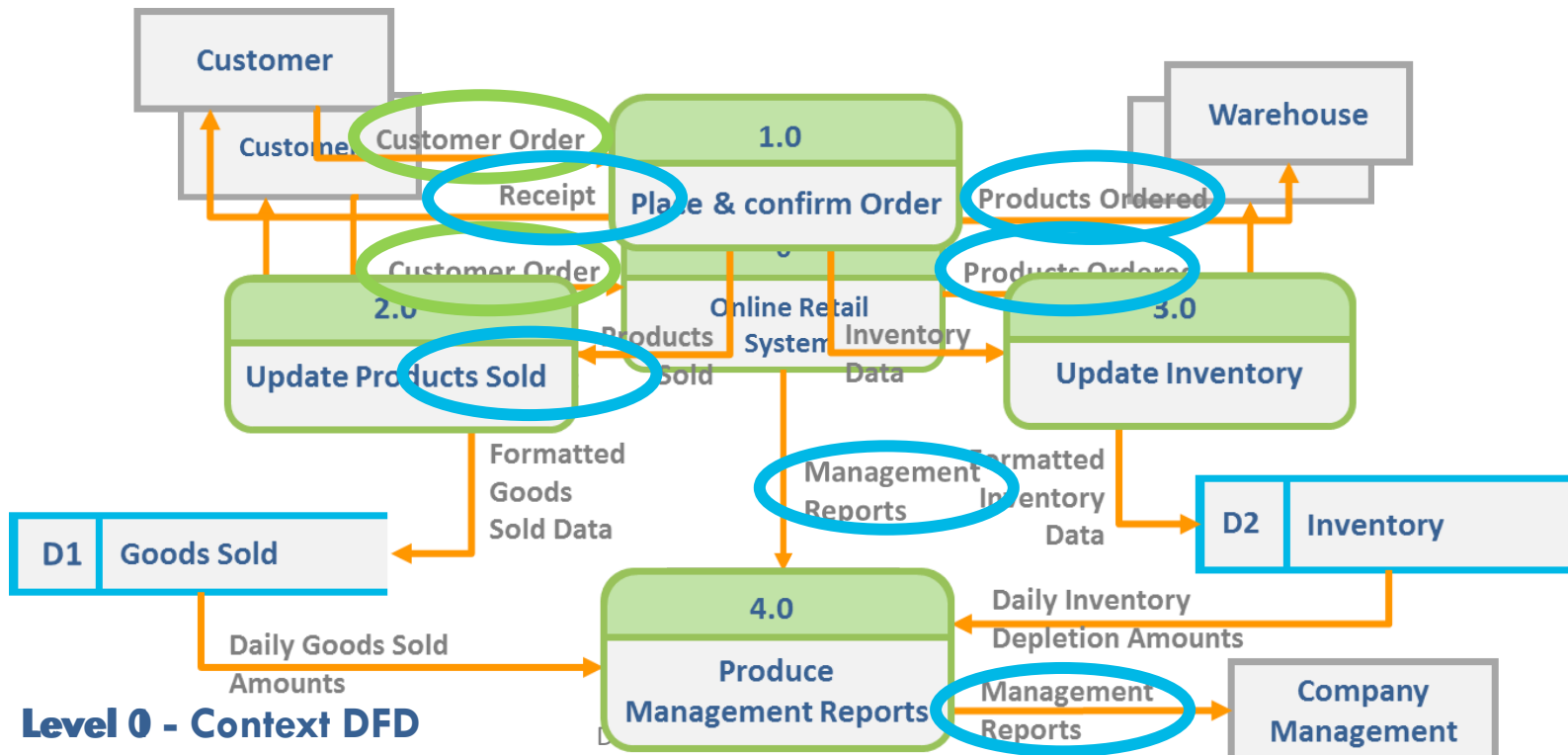


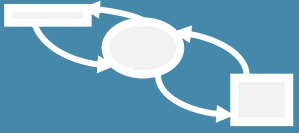


DFD

balancing

- Στα διαγράμματα DFD, όταν «κατεβαίνουμε» από ένα επίπεδο αποδόμησης (π.χ. από το Level 0 στο Level 1), είναι σημαντικό να διατηρείται η ισορροπία των ροών δεδομένων: δηλαδή, ο αριθμός των εισόδων και εξόδων που συνδέονται με μια διαδικασία στο ανώτερο επίπεδο πρέπει να αντανακλάται με ακρίβεια στις διαδικασίες του επόμενου επιπέδου. **Αυτό ονομάζεται balancing**

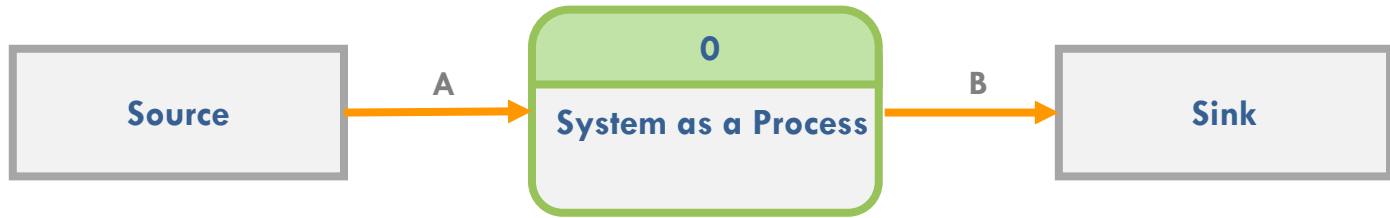




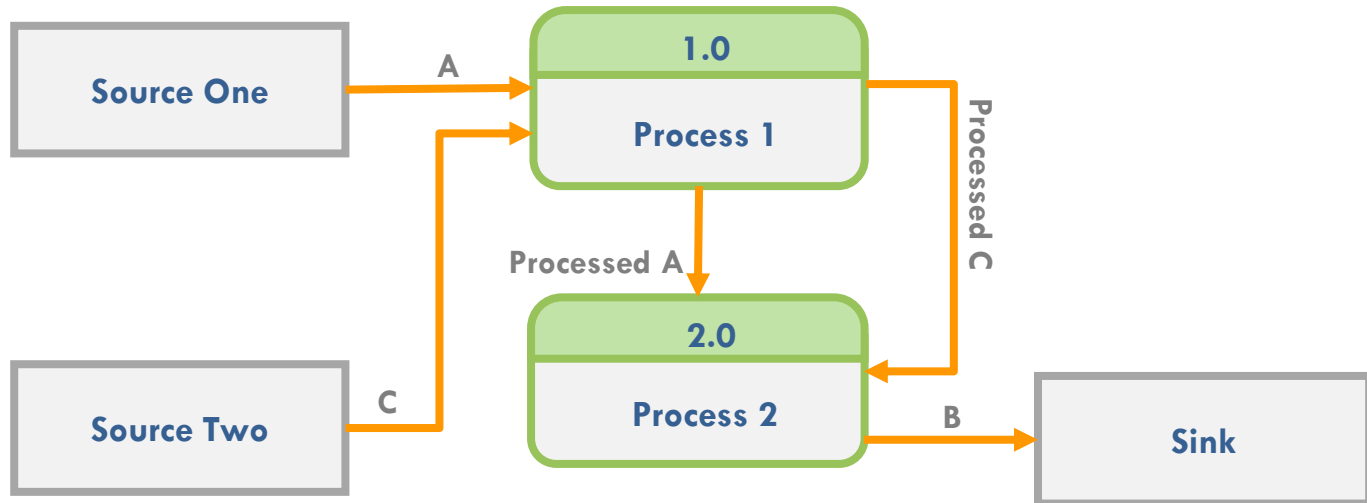
DFD Balancing

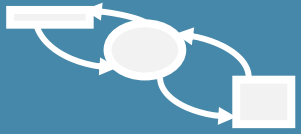
unbalanced example

Context Diagram

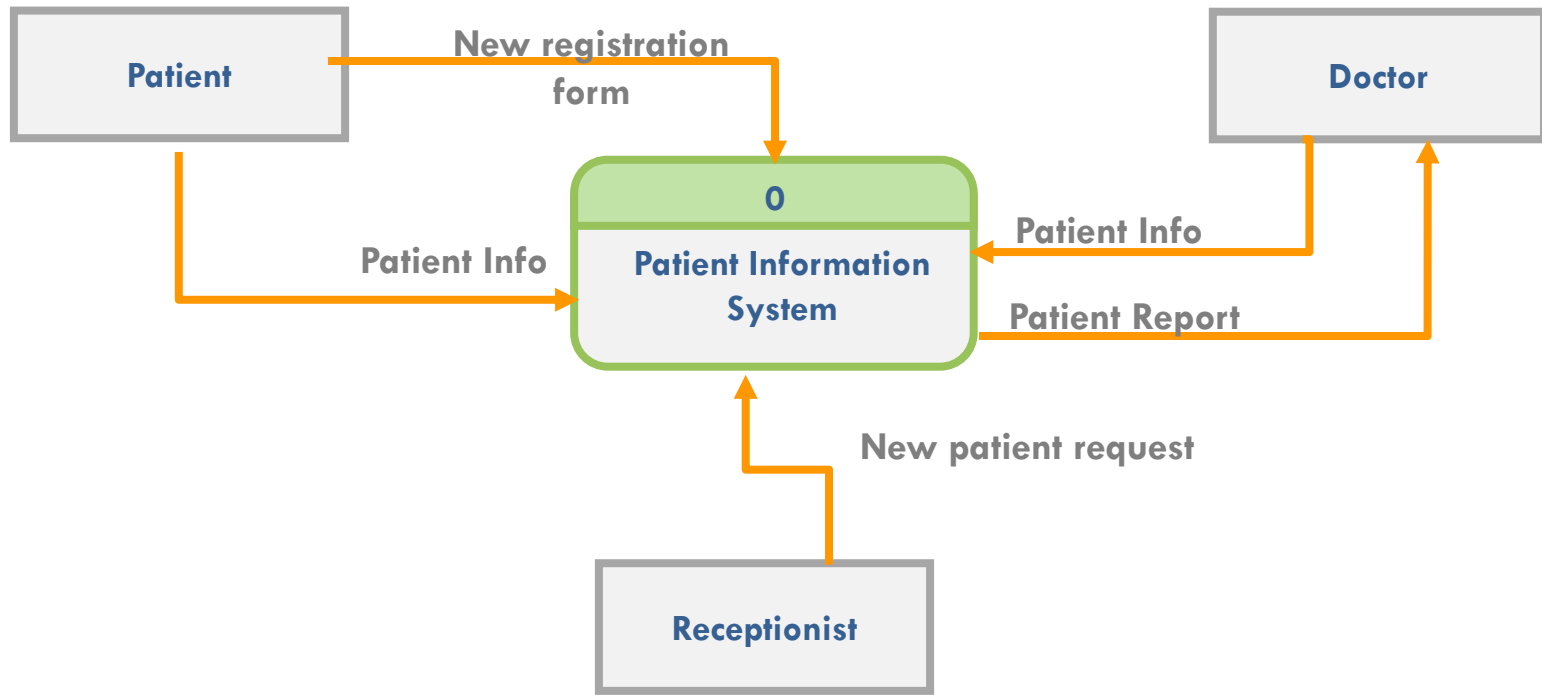


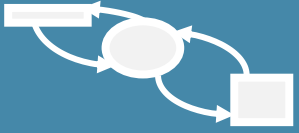
Level 1





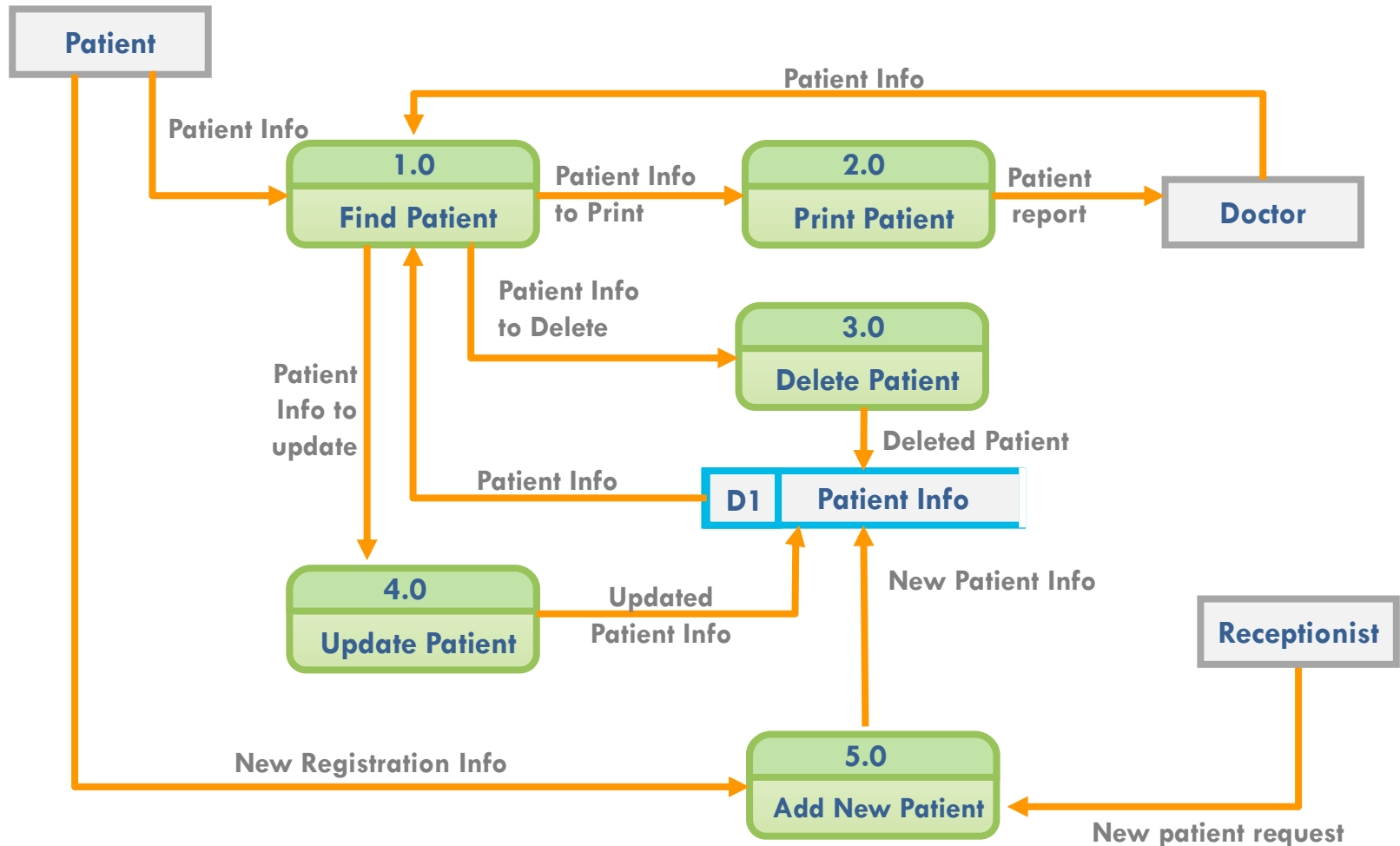
Example 2: Patient Information System Context Diagram





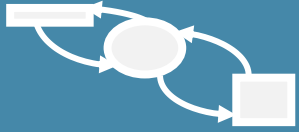
Example 2: Patient Information System

Level 1 DFD



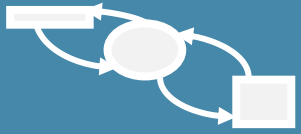


BREAKOUT SESSION



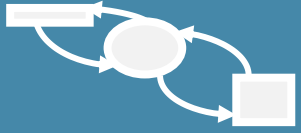
Example Case study

- Student submits registration form
- Student details are stored in the student file
- Student file is processed to generate class lists which are stored in class list files.
- Class lists are sent to tutors
- Tutors submit grades
- Results are calculated
- Results are stored on the student file
- Results are sent to students



Context Diagram

At the top level: external entities and the system



Level 1



Which ones are not correct?



Logical and Physical Data Flow Diagrams

- **Logical**

- εστιάζουν στο τι κάνει το σύστημα από επιχειρησιακή σκοπιά, περιγράφοντας τις δραστηριότητες που λαμβάνουν χώρα και τα δεδομένα που απαιτούνται ή παράγονται.
- Δεν ασχολούνται με τεχνικές λεπτομέρειες ή με το πώς θα υλοποιηθεί το σύστημα, αλλά με το πώς λειτουργεί η επιχείρηση.
- Ιδανικά για ανάλυση απαιτήσεων και κατανόηση της ροής εργασιών ανεξαρτήτως τεχνολογίας.

- **Physical**

- περιγράφουν πώς θα υλοποιηθεί το σύστημα, απεικονίζοντας τεχνικά στοιχεία, όπως βάσεις δεδομένων, διεπαφές, χρήστες, μορφές εισόδου/εξόδου
- Απαντούν στο πως



Features Common of Logical and Physical Data Flow Diagrams

Χαρακτηριστικό Σχεδίασης

Τι απεικονίζει το μοντέλο

Τι αντιπροσωπεύουν οι διαδικασίες

Τι αντιπροσωπεύουν οι αποθήκες

Τύπος αποθηκών δεδομένων

Έλεγχοι συστήματος

Λογικό

Πώς λειτουργεί η επιχείρηση

Επιχειρησιακές δραστηριότητες

Συλλογές δεδομένων ανεξαρτήτως τρόπου αποθήκευσης

Μόνιμες συλλογές δεδομένων

Επιχειρησιακοί έλεγχοι

Φυσικό

Πώς θα υλοποιηθεί το σύστημα (ή πώς λειτουργεί το υπάρχον σύστημα)

Προγράμματα, αρθρωτά μέρη λογισμικού, και χειροκίνητες διαδικασίες

Φυσικά αρχεία και βάσεις δεδομένων, χειροκίνητα αρχεία

Κύρια αρχεία, αρχεία μετάβασης. Κάθε διαδικασία που εκτελείται σε διαφορετικές χρονικές στιγμές πρέπει να συνδέεται με αποθήκη δεδομένων

Έλεγχοι εισόδου, καταγραφή κατάστασης, έλεγχος επιτυχούς ολοκλήρωσης, ασφάλεια συστήματος (π.χ. αρχεία καταγραφής)



The Progression of Models from Logical to Physical

Current Logical
Data Flow Diagram



New Logical
Data Flow Diagram



New Physical
Data Flow Diagram

Derive the logical DFD for the current system by examining the physical DFD and isolating unique business activities.

Create the logical DFD for the new system by adding the input, output, and processes required in the new system to the logical DFD for the current system.

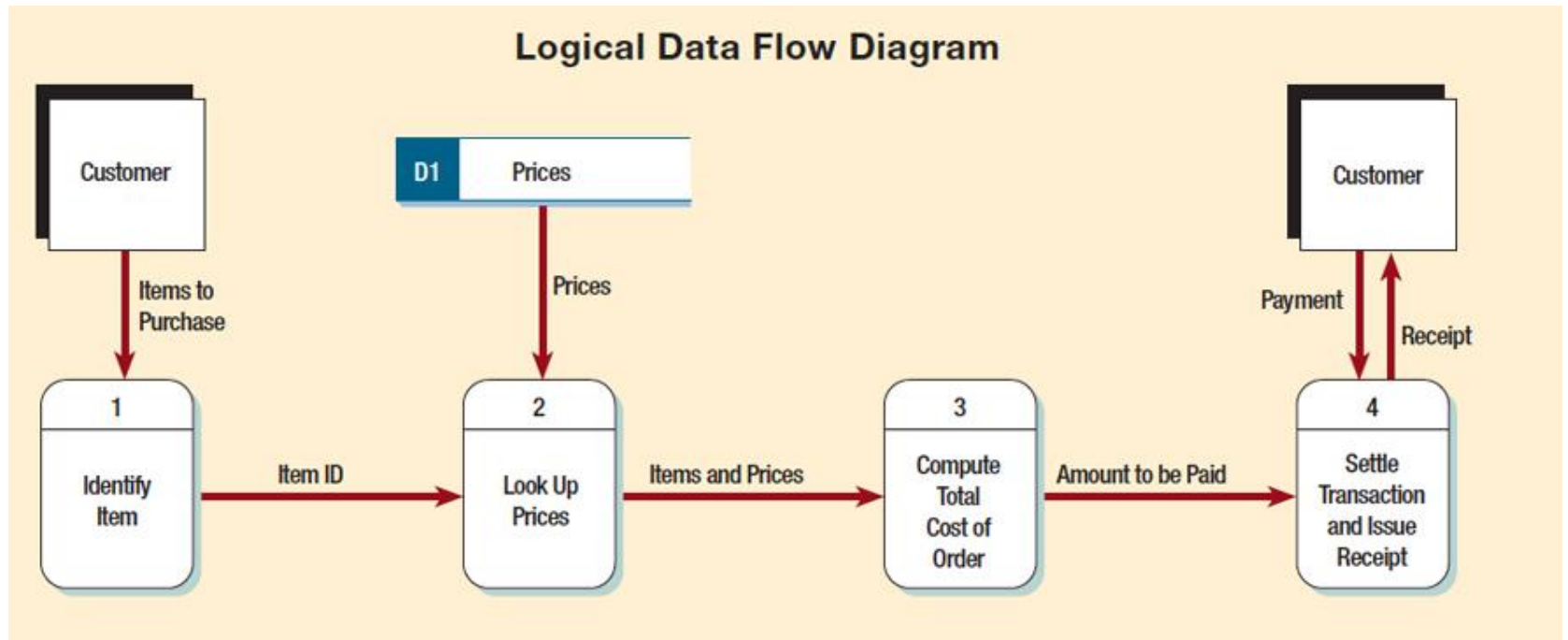
Derive the physical DFD by examining processes on the new logical diagram. Determine where the user interfaces should exist, the nature of the processes, and necessary data stores.

Η εξέλιξη της δημιουργίας διαγραμμάτων ροής δεδομένων (DFD) περιλαμβάνει τα εξής στάδια:

- **Ανάλυση του υφιστάμενου συστήματος** (τρέχον λογικό DFD).
- **Προσθήκη χαρακτηριστικών που θα πρέπει να περιλαμβάνει το νέο σύστημα** (προτεινόμενο λογικό DFD).
- **Ανάπτυξη των βέλτιστων μεθόδων υλοποίησης του νέου συστήματος** (φυσικό DFD).

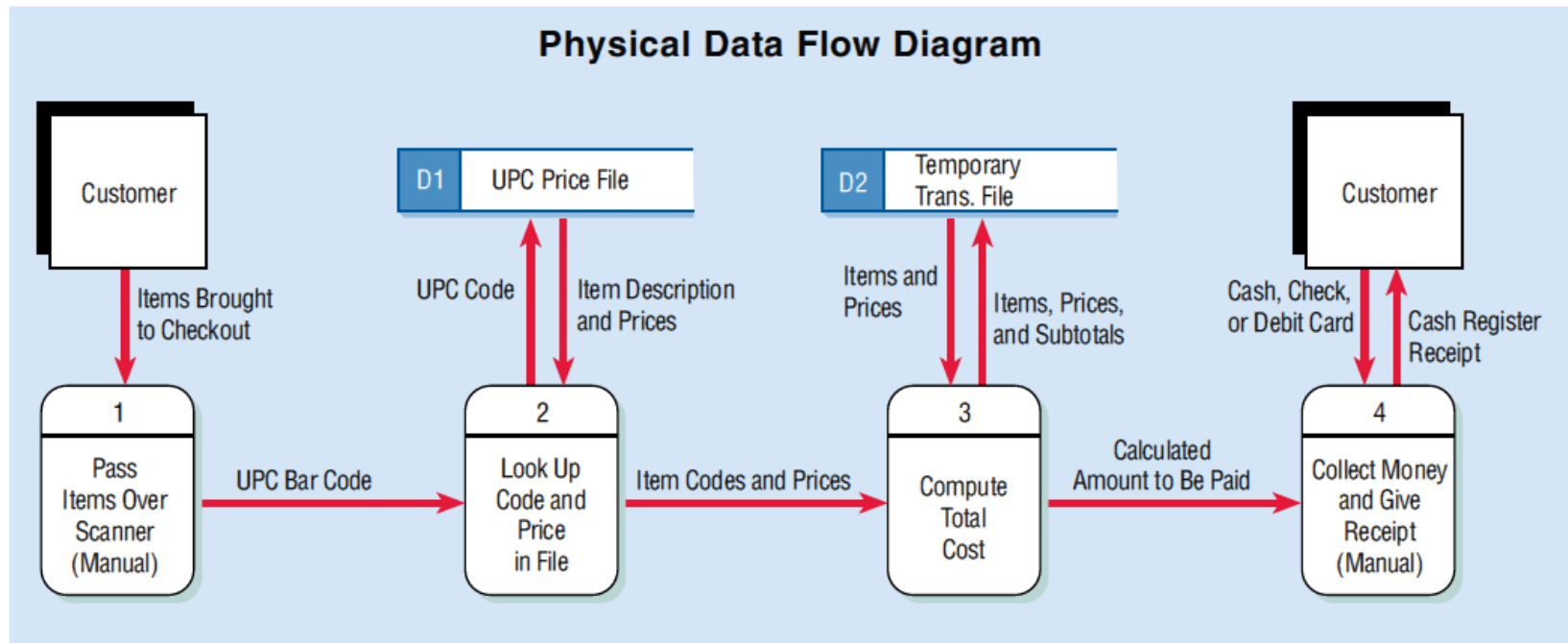


Logical DFD Example





Physical DFD Example





Developing Logical Data Flow Diagrams

- Better communication with users
- More stable systems
- Better understanding of the business by analysts
- Flexibility and maintenance
- Elimination of redundancy and easier creation of the physical model



Developing Physical Data Flow Diagrams

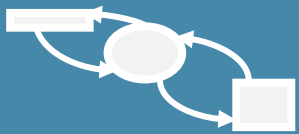
- Clarifying which processes are performed by humans and which are automated
- Describing processes in more detail
- Sequencing processes that have to be done in a particular order
- Identifying temporary data stores
- Specifying actual names of files and printouts
- Adding controls to ensure the processes are done properly



Physical Data Flow Diagrams Contain Many Items Not Found in Logical Data Flow Diagrams

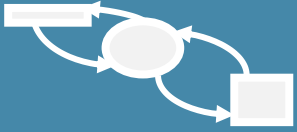
Περιεχόμενα των Φυσικών Διαγραμμάτων Ροής Δεδομένων:

- Χειροκίνητες διαδικασίες
- Διαδικασίες για προσθήκη, διαγραφή, τροποποίηση και ενημέρωση εγγραφών
- Εισαγωγή και επαλήθευση δεδομένων
- Διαδικασίες επικύρωσης για διασφάλιση της ορθότητας των εισαγόμενων δεδομένων
- Ακολουθίες διαδικασιών για αναδιάταξη της σειράς των εγγραφών
- Διαδικασίες για παραγωγή κάθε μοναδικής εξόδου του συστήματος
- Ενδιάμεσες αποθήκες δεδομένων
- Πραγματικά ονόματα αρχείων που χρησιμοποιούνται για αποθήκευση δεδομένων
- Έλεγχοι για την επιβεβαίωση ολοκλήρωσης ενεργειών ή την ύπαρξη λαθών



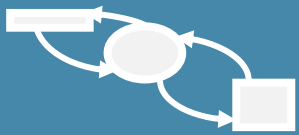
Event Modeling and Data Flow Diagrams

- Μια εισερχόμενη ροή από μια εξωτερική οντότητα ονομάζεται μερικές φορές ενεργοποιητής (trigger) επειδή εκκινεί τις δραστηριότητες μιας διαδικασίας.
- Τα συμβάντα προκαλούν το σύστημα να εκτελέσει κάποια ενέργεια και λειτουργούν ως ενεργοποιητές για το σύστημα.
- Μία προσέγγιση για τη δημιουργία φυσικών διαγραμμάτων ροής δεδομένων είναι να δημιουργείται ένα απόσπασμα DFD για κάθε μοναδικό συμβάν του συστήματος



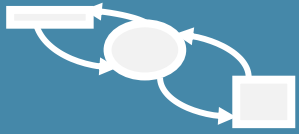
Event response tables

- Ένας πίνακας συμβάντων χρησιμοποιείται για τη δημιουργία ενός διαγράμματος ροής δεδομένων (DFD) μέσω της ανάλυσης κάθε συμβάντος και των δεδομένων που χρησιμοποιούνται και παράγονται από αυτό.
- Κάθε γραμμή στον πίνακα συμβάντων αντιστοιχεί σε ένα απόσπασμα διαγράμματος ροής δεδομένων και χρησιμοποιείται για να δημιουργηθεί μια μεμονωμένη διαδικασία στο διάγραμμα.

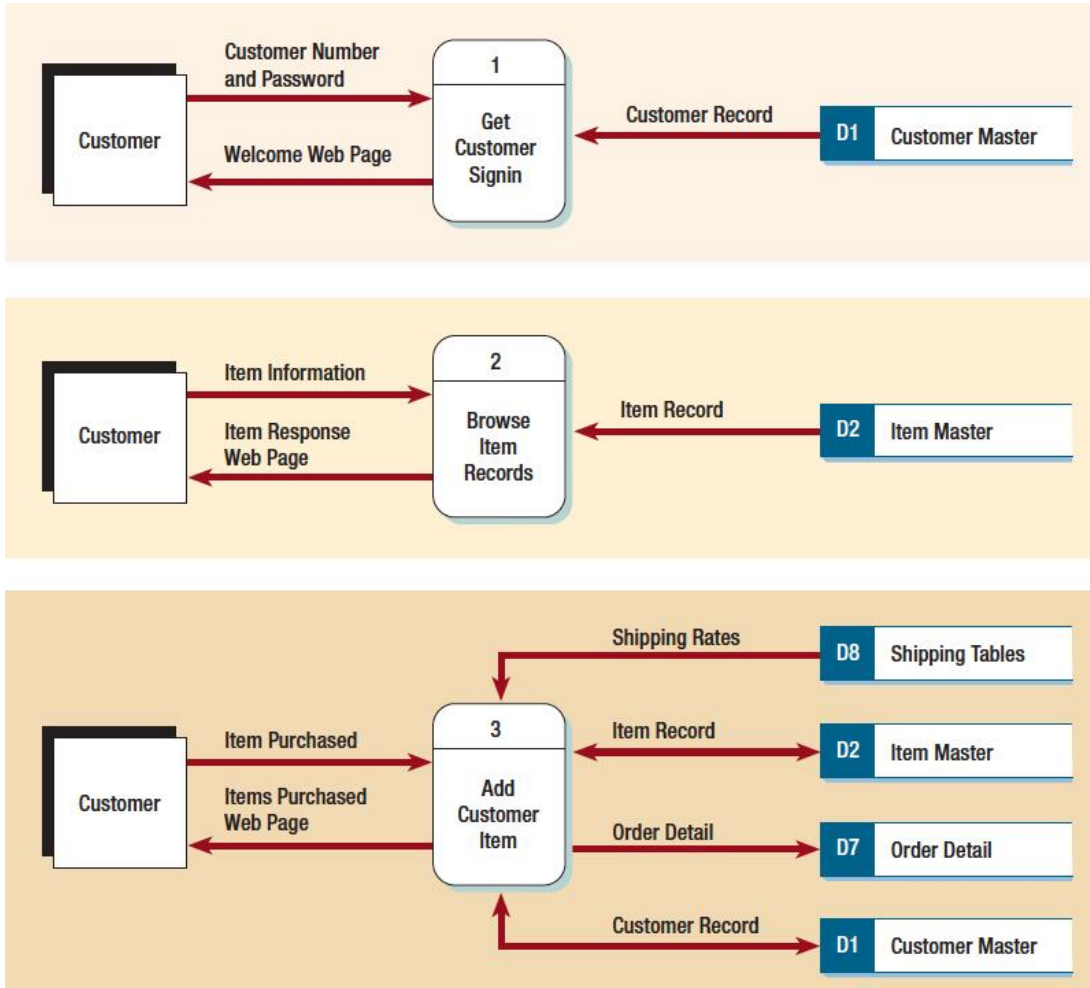


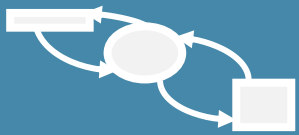
An Event Response Table for an Internet Storefront

Event	Source	Trigger	Activity	Response	Destination
Customer logs on	Customer	Customer number and password	Find customer record and verify password. Send Welcome web page	Welcome web page	Customer
Customer browses items at Web storefront	Customer	Item information	Find item price and quantity available. Send Item Response web page.	Item Response web page	Customer
Customer places item into shopping basket at Web storefront	Customer	Item purchase (item number and quantity)	Store data on Order Detail Record. Calculate shipping cost using shipping tables. Update customer total. Update item quantity on hand.	Items Purchased web page	Customer
Customer checks out	Customer	Clicks “Check Out” button on web page	Display Customer Order web page.	Verification web page	
Obtain customer Payment	Customer	Credit card information	Verify credit card amount with credit card company. Send.	Credit card data Customer feedback	Credit card company Customer
Send customer email		Temporal, hourly	Send customer an email confirming shipment.		Customer



Data Flow Diagrams for the First Three Rows of the Internet Storefront Event Response Table





Use Cases and Data Flow Diagrams

- Κάθε use case ορίζει μία δραστηριότητα και το αντίστοιχο trigger (δηλαδή το συμβάν που την ενεργοποιεί), την είσοδο και την έξοδο της.
- Αυτό επιτρέπει στον αναλυτή να συνεργαστεί με τους χρήστες ώστε να κατανοήσει τη φύση των διαδικασιών και των δραστηριοτήτων και στη συνέχεια να δημιουργήσει ένα απόσπασμα διαγράμματος ροής δεδομένων (data flow diagram fragment).



Recap

learning check

- Structured Systems Analysis and Design approach is another way of specifying requirements
- Discussed one of SSAD's modelling technique and its associated diagrams:
 - Process or Functional Modelling (DFDs)
 - Context-level data flow diagram
 - Level 1 logical data flow diagram
 - Level 2...n Child diagrams
 - Creating the Logical DFD (Analysis)
 - Create from the logical Data Flow diagram
 - A step closer to programming
 - Creating the Physical DFD (Design)
 - Create from the logical Data Flow diagram
 - A step closer to programming
 - Next week: Data Modelling (Entity Relationship Diagrams) and Entity Life History



Further Reading

- Kendall and Kendall (2023) Systems analysis and design – Chapter 7, 8, 9