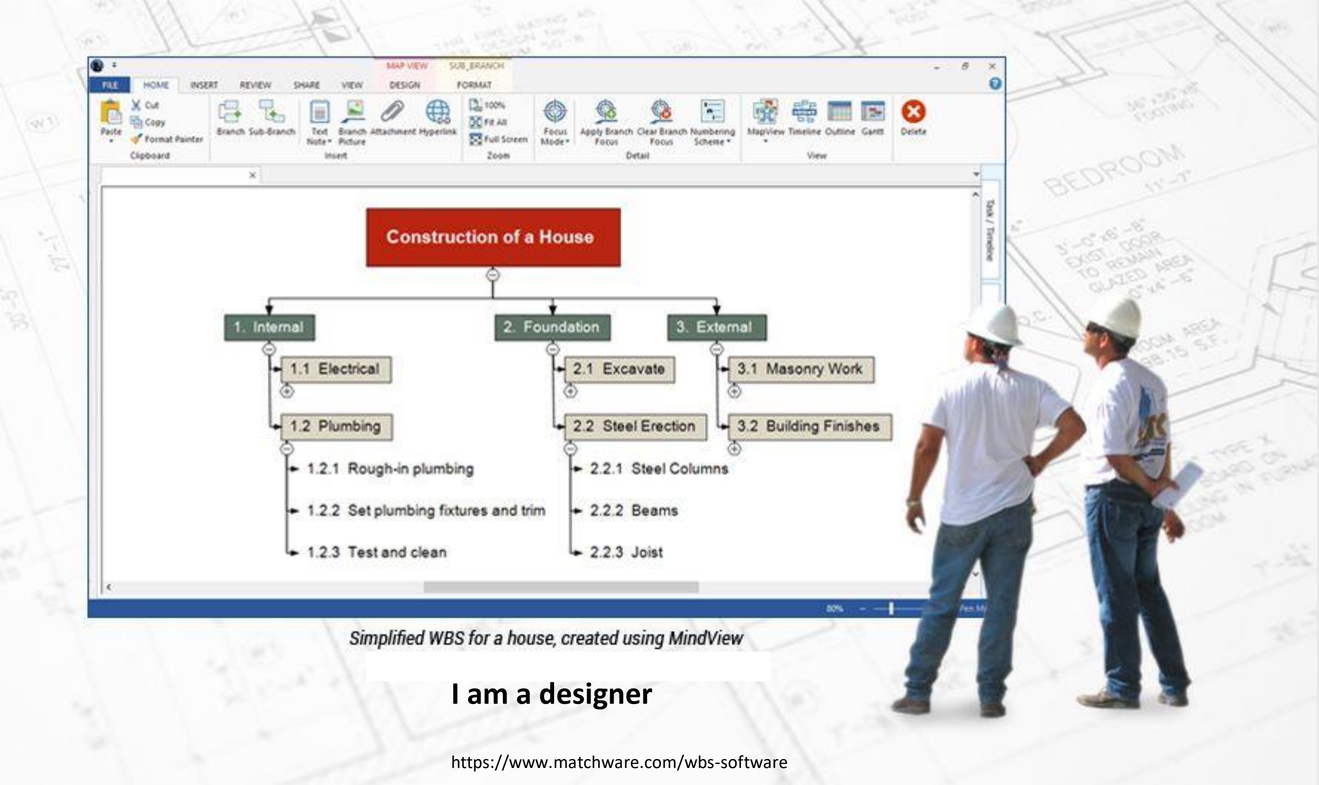
Week 8: System Structure:
Fundamental Modelling Concepts (FMC)

IFB103, QUT, Dr Alireza Nili



Simplified WBS for a house, created using MindView

I am a designer

https://www.matchware.com/wbs-software

Week 8 assessment structure

Before your week 8 tutorial:

1. Ensure you have watched Week 8 lecture recording.
2. Read the assessment specification file related to Design Challenge 2 (the final assessment, available on Blackboard).

During week 8:

1. Read the Week 8 tutorial slides. Think on items 1, 2a and 2b in the assessment specification file. Draw the first draft of your requirements matrix (item 1) and FMC diagram (item 2a) and answer item 2b. [Assessed items]

2. Show the first draft of your requirements matrix and FMC diagram to your tutor during your tutorial. Your tutor will give you feedback on your work during the tutorial.

This is what you need to do every week:

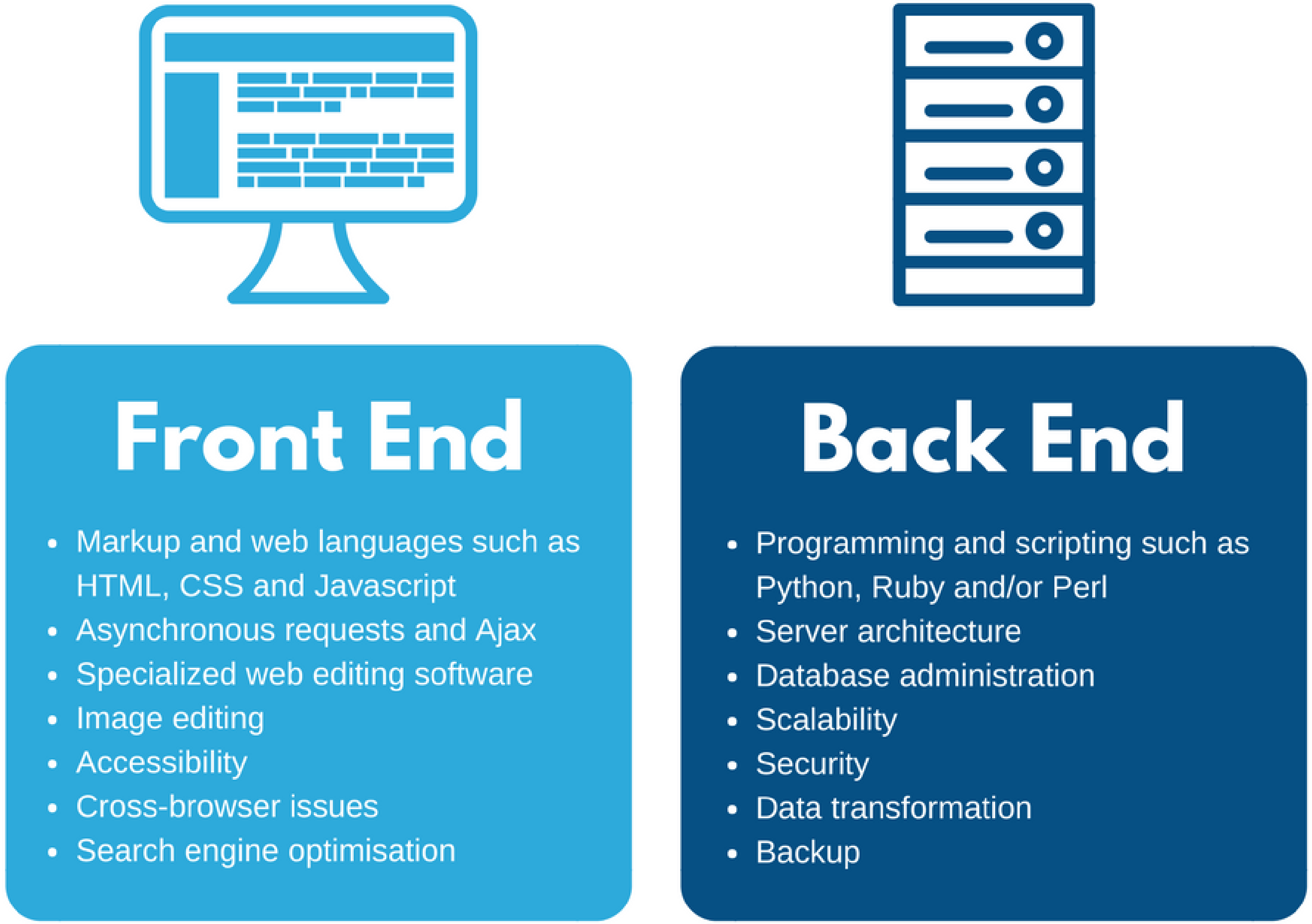
1. Read/review the relevant parts of the assessment specification file (Design Challenge 2) that we have posted on Blackboard (by relevant parts, we mean parts of the file that are relevant to the current week's tasks).
2. Watch the lecture recording.
3. Read the tutorial slides and other teaching resources each week 'before' attending your tutorial.
4. You need to draw the first draft of your diagram before attending your tutorial and show it to your tutor during the session.
5. Your tutor will answer your questions and provides feedback for improving your diagram.

The above approach gives you an extended time for reading the teaching material and an opportunity for a personalised consultation for receiving feedback on your work during the tutorial.

From DC1 to DC2

We have so far been working on prototyping, user experience and front end design.

We will now shift to understanding more about systems analysis and design.



Front End

- Markup and web languages such as HTML, CSS and Javascript
- Asynchronous requests and Ajax
- Specialized web editing software
- Image editing
- Accessibility
- Cross-browser issues
- Search engine optimisation

Back End

- Programming and scripting such as Python, Ruby and/or Perl
- Server architecture
- Database administration
- Scalability
- Security
- Data transformation
- Backup

Goals for today

- Requirements determination and analysis
- Understand the concept of system structure
- Understand the principles of FMC.

- An illustrative example of FMC.

Background

Requirements Determination and Analysis

Elements of Block Diagrams and an Example

Recommendations

Wrap up

Background

- Building software-intensive systems is often a complex process.

- It requires the technical means to support the creation process, and a high degree of efficiency to communicate analysis and design ideas.

- Without this efficiency in communication it would hardly be possibly to plan how the system components work together.

- This applies to every analyst, designer, developer, manager and team leader.

“The business of software building […is] most of all a business of talking to each other and writing things down. Those who were making major contributions to the field were more likely to be its best communicators than its best technicians.”

Tom DeMarco [Why does software cost so much?]

There are two main steps of analysis & design for the back-end of the system

Two “main” steps:

1) Requirements determination

2) Visualising the results of system analysis via software such as Visual Paradigm, Visio and LucidChart

Background

Requirements Determination and Analysis

Elements of Block Diagrams and an Example

Recommendations

Wrap Up

Step 1: Requirements Determination

(requirements determination/elicitation & analysis)

Important:

We focus on **……..???……..** type of requirements.

Why this type of requirement is particularly important for systems analysis and design?

Item 1 – Requirements Table: guidelines

- You need to read Case 1 and record the requirements in the template that we have provided for you. The template is available in the relevant tutorial slides and as a separate file on Blackboard.

- Each requirement should be concise (usually between two to five words) and should start with a verb (e.g. “search purchase history”, “pay online”, and “provide…”).

- In each row of the table, you need to mention whether a requirement is essential or desirable. Essential means that the system won’t work without that requirement. Desirable means that the system is able to work without that requirement, but it would be good to have it implemented.

- Show the first draft/version of your table to your tutor during the tutorial to receive feedback.

Step 2: Visualising the Results

This week we use Fundamental Modelling Concepts (FMC) to visualise the next week's analysis.

Next, we will use Unified Modelling Language (UML) to visualise our analysis of a system more deeply in the next few weeks.

- A **FMC** diagram is a graphical description of a software-intensive system.

- **FMC diagrams** help to depict the compositional structure of any system that processes information.

- The goal is to capture the essential structures necessary to understand a system behaviour (internal and to its environment) and to describe these structures in a comprehensive way.

- Using FMC, designers can visualize didactic models (i.e. models that enable people to share a common understanding of a system's structure). Therefore FMC helps to reduce cost and risk in the design and handling of complex systems.

Three aspects of FMC diagrams

FMC distinguishes three fundamental aspects namely the compositional structure, the dynamic structure and the value range structure.

1. **Block diagrams depict compositional structure** [this is the focus of this lecture].
2. Petri nets depict dynamic structure.
3. Entity-Relationship diagrams depict value range structure.

Note: a FMC metamodel distinguishes all three types of structures.

http://www.fmc-modeling.org/metamodel

Background

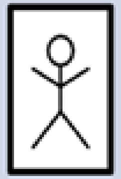
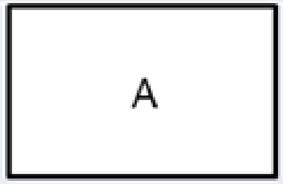
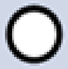
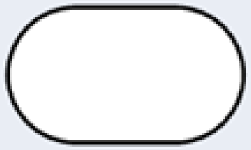
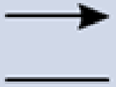
Requirements Determination and Analysis

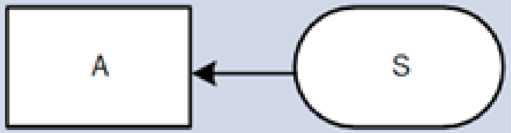
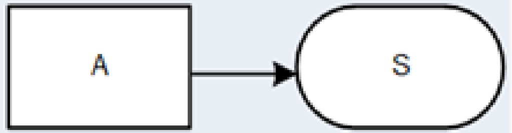
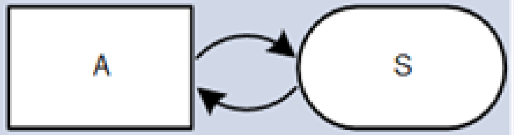
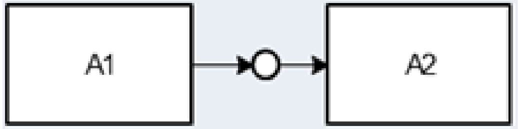
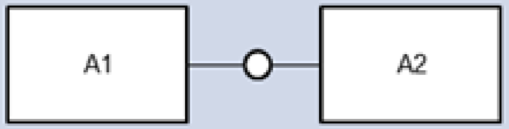
Elements of Block Diagrams and an Example

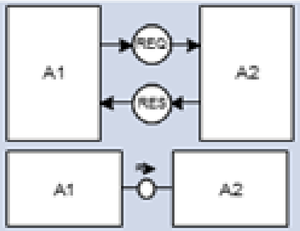
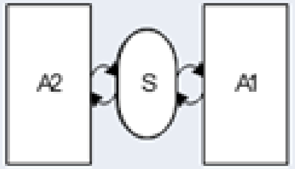
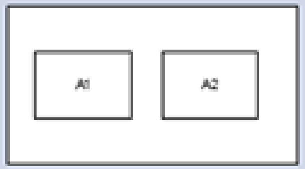
Recommendations

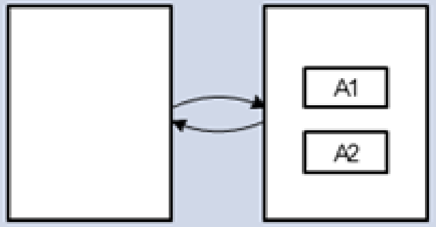
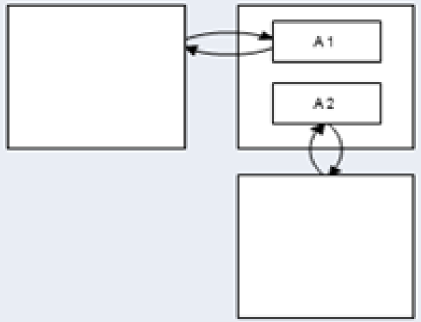
Wrap up

Elements of block diagrams

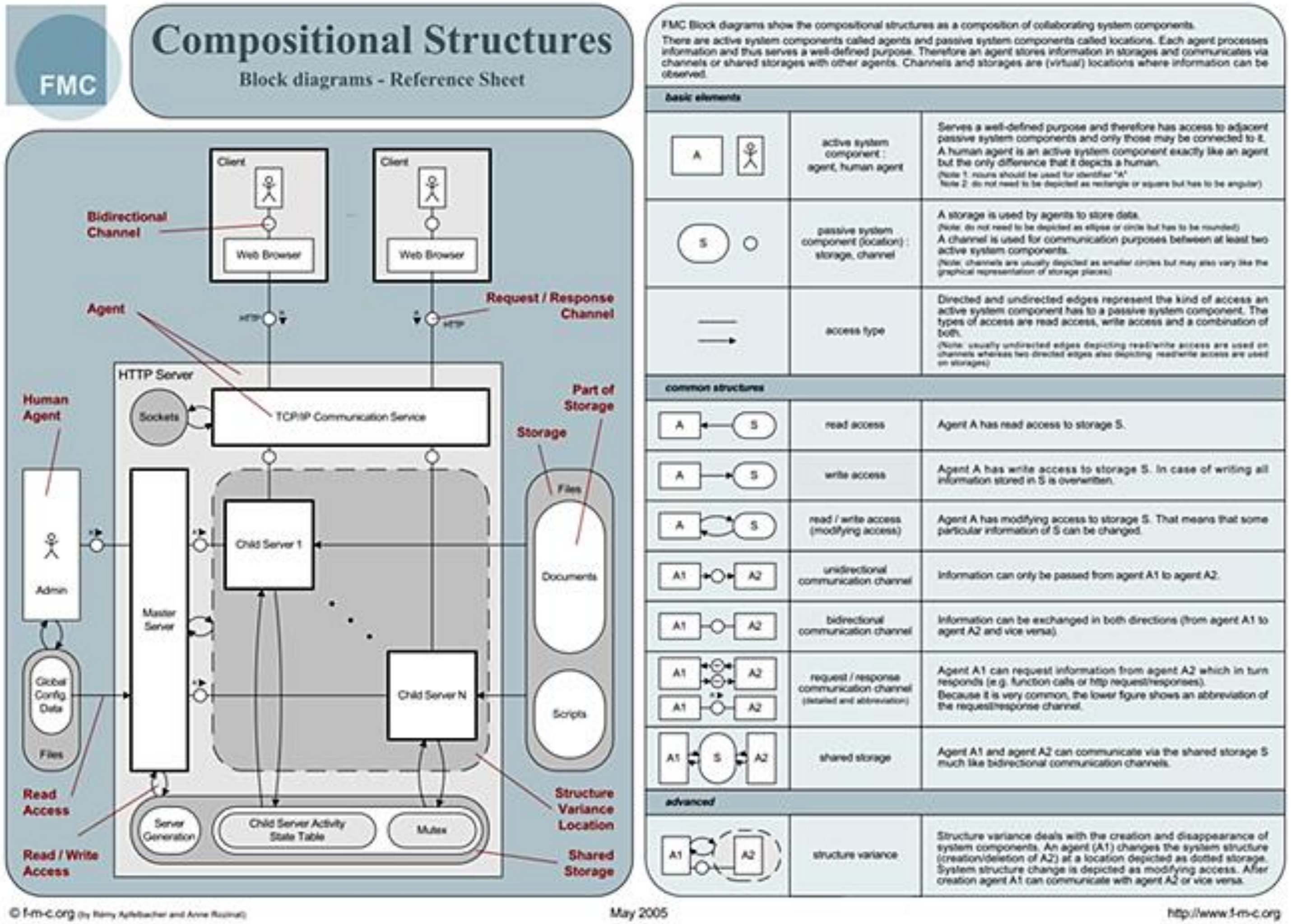
Symbol	Meaning
	Human role (type of active component)
	System component (type of active component)
	Access point or channel (type of passive component)
	Storage (type of passive component)
	Interaction

Symbol	Meaning
	Read access from storage by active component (note human role or systems component)
	Write access to storage by active component
	Read/write access from/to storage by active component
	Unidirectional communication between active components
	Bidirectional communication between active components

Symbol	Meaning
	Request/response between active components
	Shared storage between active components
	Decomposition of systems components

Symbol	Meaning
	General read/write access with composite systems component
	Specialised read/write access with composite systems component

Available on Blackboard:



Compositional Structures

Block diagrams - Reference Sheet

FMC Block diagrams show the compositional structures as a composition of collaborating system components.

There are active system components called agents and passive system components called locations. Each agent processes information and thus serves a well-defined purpose. Therefore an agent stores information in storages and communicates via channels or shared storages with other agents. Channels and storages are (virtual) locations where information can be observed.

basic elements		
A	active system component : agent, human agent	Serves a well-defined purpose and therefore has access to adjacent passive system components and only those may be connected to it. A human agent is an active system component exactly like an agent but the only difference that it depicts a human. (Note 1: nouns should be used for identifier "A" Note 2: do not need to be depicted as rectangle or square but has to be angular)
S	passive system component (location) : storage, channel	A storage is used by agents to store data. (Note: do not need to be depicted as ellipse or circle but has to be rounded) A channel is used for communication purposes between at least two active system components. (Note: channels are usually depicted as smaller circles but may also vary like the graphical representation of storage places)
	access type	Directed and undirected edges represent the kind of access an active system component has to a passive system component. The types of access are read access, write access and a combination of both. (Note: usually undirected edges depicting read/write access are used on channels whereas two directed edges also depicting read/write access are used on storages)
common structures		
A S	read access	Agent A has read access to storage S.
A S	write access	Agent A has write access to storage S. In case of writing all information stored in S is overwritten.
A S	read / write access (modifying access)	Agent A has modifying access to storage S. That means that some particular information of S can be changed.
A1 A2	unidirectional communication channel	Information can only be passed from agent A1 to agent A2.
A1 A2	bidirectional communication channel	Information can be exchanged in both directions (from agent A1 to agent A2 and vice versa).
A1 A2 A1 A2	request / response communication channel (detailed and abbreviation)	Agent A1 can request information from agent A2 which in turn responds (e.g. function calls or http request/responses). Because it is very common, the lower figure shows an abbreviation of the request/response channel.
A1 S A2	shared storage	Agent A1 and agent A2 can communicate via the shared storage S much like bidirectional communication channels.
advanced		
A1 A2	structure variance	Structure variance deals with the creation and disappearance of system components. An agent (A1) changes the system structure (creation/deletion of A2) at a location depicted as dotted storage. System structure change is depicted as modifying access. After creation agent A1 can communicate with agent A2 or vice versa.

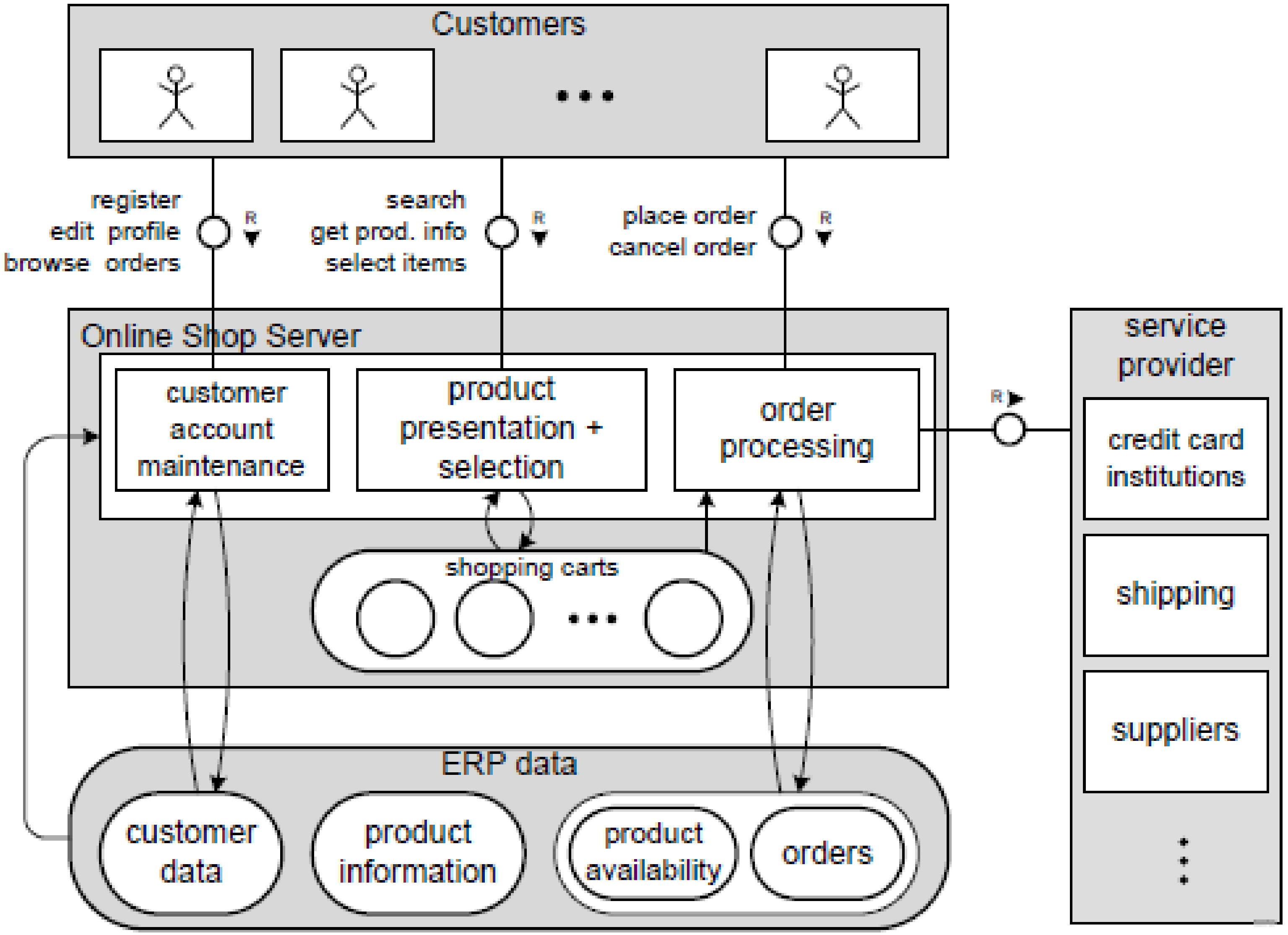
© f-m-c.org (by Rémy Apfelbacher and Anne Rozinat)

May 2005

http://www.f-m-c.org

Accessed on 12 Aug 2018 from: http://www.fmc-modeling.org/download/notation_reference/Reference_Sheet-Block_Diagram.pdf

A simple block diagram showing the compositional structure of an online shop



To process an order, the online shop server uses **external services**, such as credit card confirmation or transport.

Reference (accessed on 12 Aug 2018) from: https://blogs.sap.com/2008/02/19/how-to-communicate-architecture-technical-architecture-modeling-at-sap-part-2/

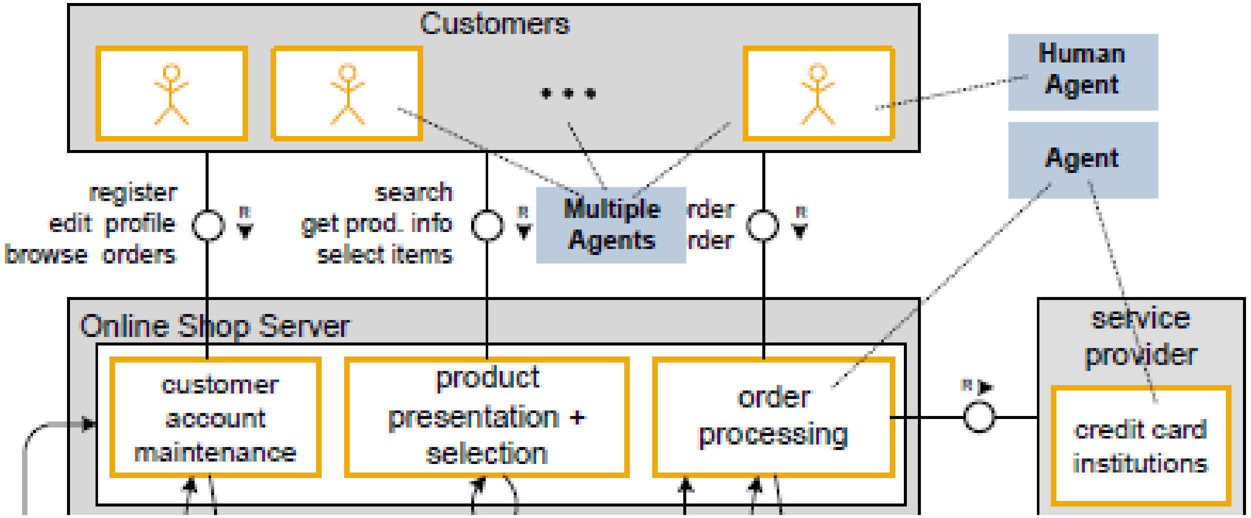
This example presents a top-level view of a system realized by:

- a network of hundreds of humans
- hundreds of computers
- software built from millions of lines of code.

It would hardly be possible to efficiently develop such a system without efficient ways to communicate design ideas among design team → why we use FMC.

Agents: the first element

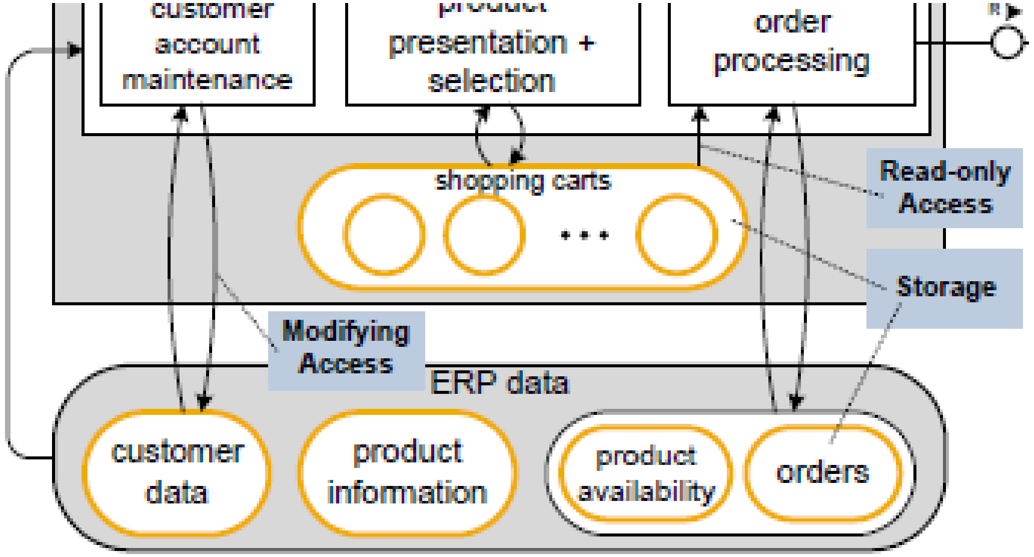
- Agents process and use information: sorting a list, read and reject a job application, calculate an average, processing tax refund, issuing an ID card, approving or rejecting a request, ...
- Agents are active components. Only agents can do something, all other elements in a block diagram are passive components.
- Agents are depicted by rectangular shapes.



Reference (accessed on 12 Aug 2018) from: https://blogs.sap.com/2008/02/19/how-to-communicate-architecture-technical-architecture-modeling-at-sap-part-2/

Storage: the second element

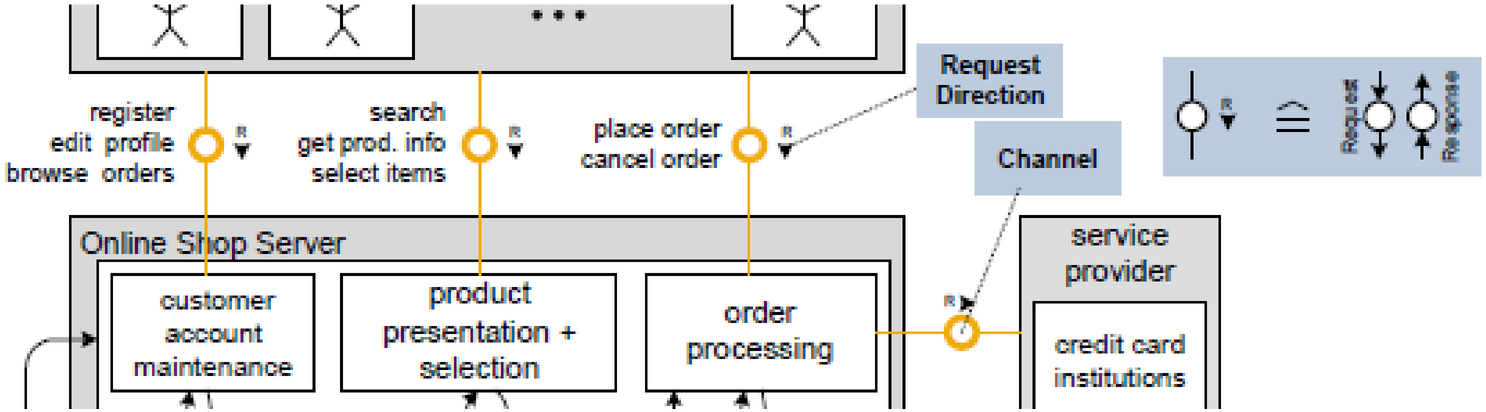
- Storage contains information that can be processed by an agent.
- Storage is a passive component – it cannot do anything on its own.
- Storage can contain any information (e.g. a tax declaration, video, an app or a mobile game.
- An agent can use a storage to store data to retrieve it later on, or to share data with other agents.
- Storage is depicted by rounded shapes.



Reference (accessed on 12 Aug 2018) from: https://blogs.sap.com/2008/02/19/how-to-communicate-architecture-technical-architecture-modeling-at-sap-part-2/

Channel: the third element

- The purpose of a channel (e.g. telephones and network connections) is to connect two or more agents, so they can communicate (e.g. to send requests, replies, messages, notifications, or calls).
- They are passive components.
- A channel is shown via a small circle on a connecting arc between agents.
- You can annotate the circle with the type of request, the type of information, or with the protocol used.
- Arrows indicate the direction of information flow.
- The "R" with the small arrow is a short notation of the request-response channel pair. In this figure, the arrow indicates the direction of the request. It is pointing from client to server.



Reference (accessed on 12 Aug 2018) from: https://blogs.sap.com/2008/02/19/how-to-communicate-architecture-technical-architecture-modeling-at-sap-part-2/

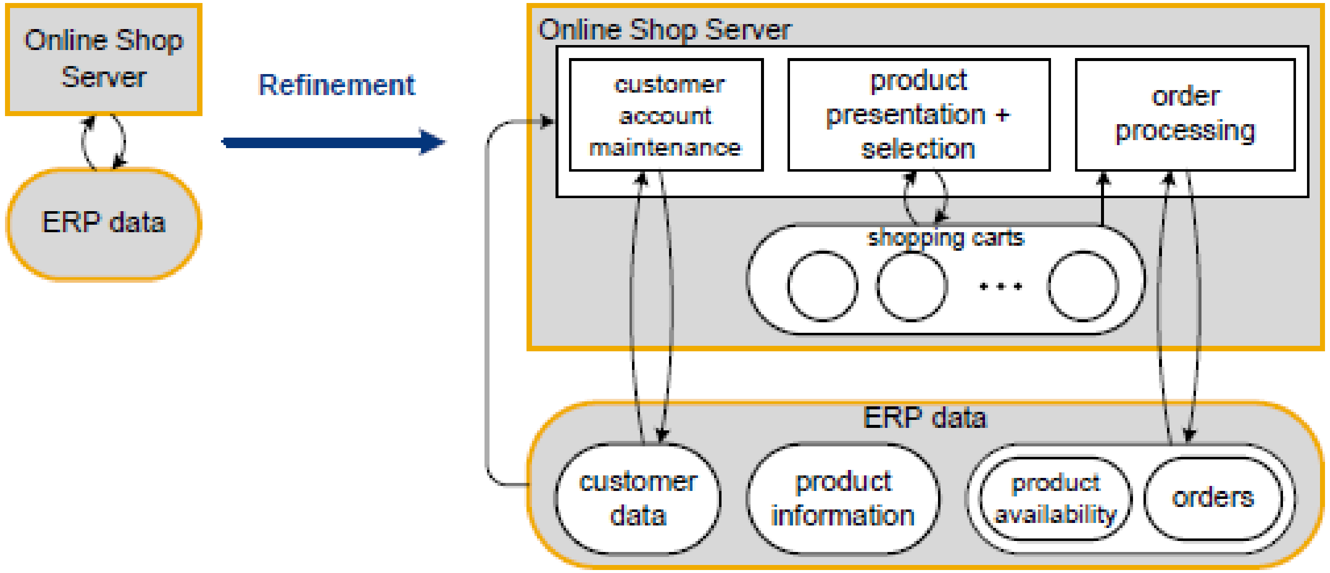
Important

Channels only connect agents with each other.

Access arcs can only be used between storage and agent.

Nesting: the fourth element

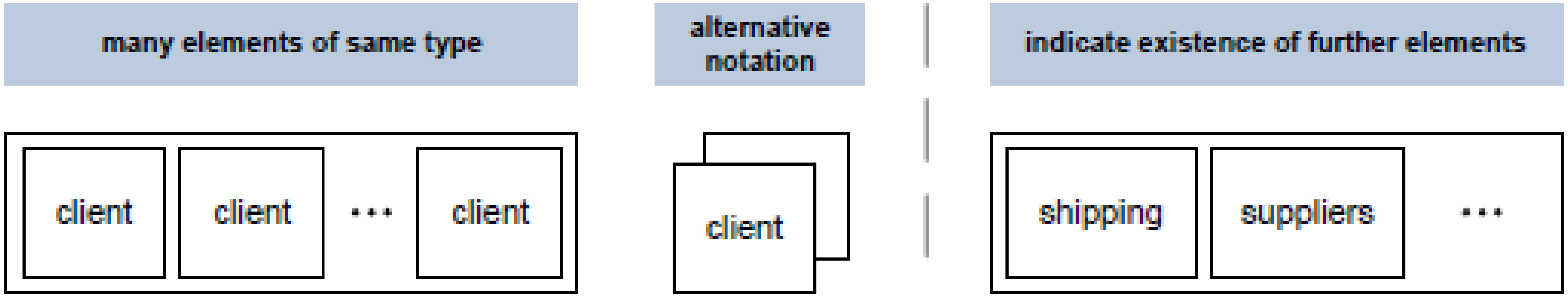
Nesting means showing the inner structure (**refining)** and providing more details about it.



Reference (accessed on 12 Aug 2018) from: https://blogs.sap.com/2008/02/19/how-to-communicate-architecture-technical-architecture-modeling-at-sap-part-2/

Multiplicity in a block diagram:

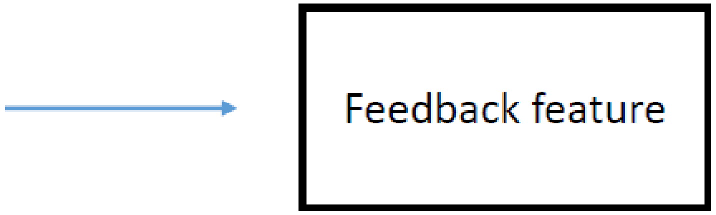
You can use two means to indicate multiplicity: Either stack the elements, or use 3 dots to indicate that there may be more.



[Reference (accessed on 12 Aug 2018) from: https://blogs.sap.com/2008/02/19/how-to-communicate-architecture-technical-architecture-modeling-at-sap-part-2/](https://blogs.sap.com/2008/02/19/how-to-communicate-architecture-technical-architecture-modeling-at-sap-part-2/)

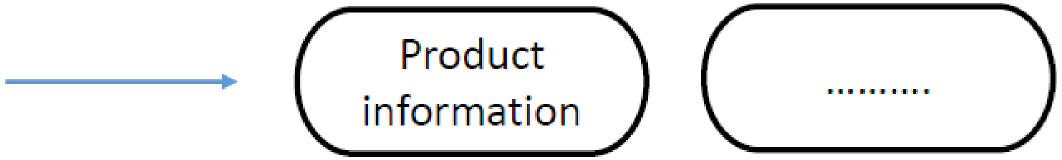
Naming/labelling an agent:

Summarise a requirement into one or two words which sound like a feature of the system. For example, the **“provide feedback for a product”** requirement in the case study should be labeled as **“feedback feature”**.



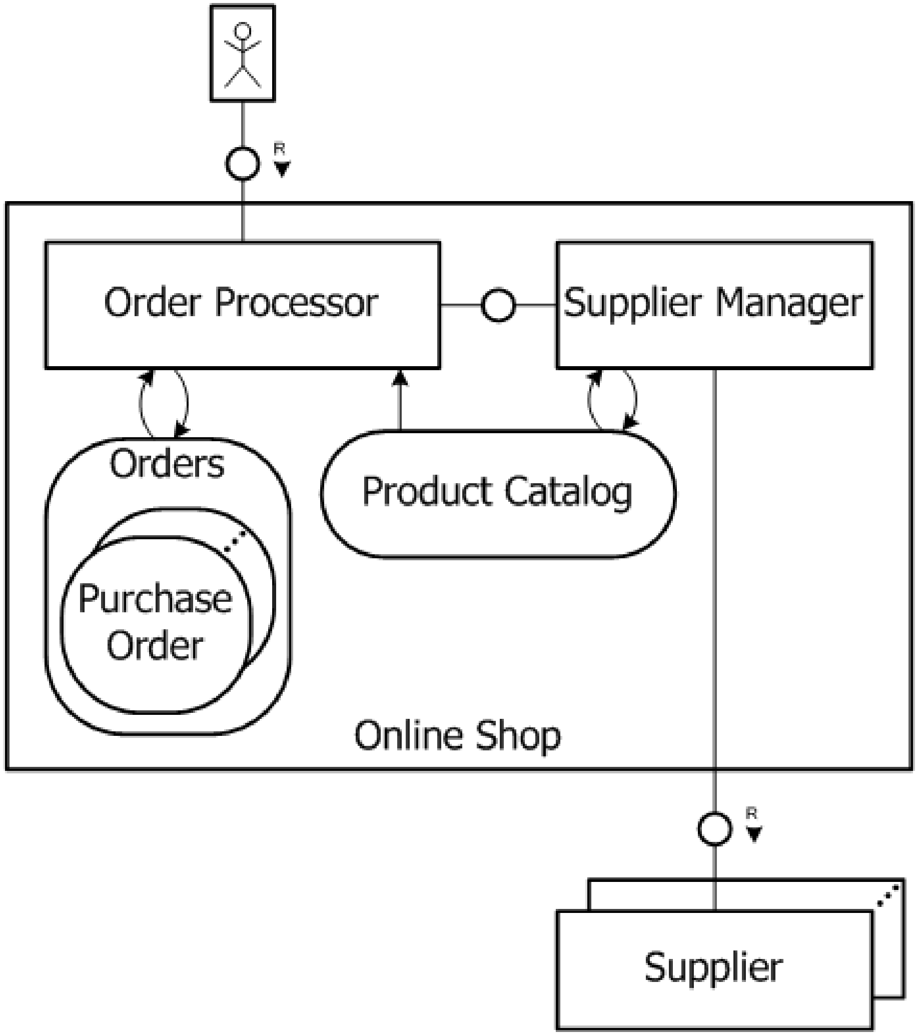
Naming/labelling storage:

The label/name should include one or two words only (e.g. **“product information”**) and need to be logically related to its agent. For example, if a customer wants to use the **“feedback feature”** to provide feedback on a product, the feedback feature should have ‘read’ access to the **“product information”** storage. Please note that you may need to connect an agent two more than one storage.



Avoid generic names, such as “agent”, “memory”, or “data”.

Analyse this FMC example. How can it be expanded?



https://commons.wikimedia.org/wiki/File:FMCBlockDiagram.png

Question:

FMC is at …………………… level, and UML is at ………………. level.

(a) Low/Design ; Top/Conceptual

(b) Top/Conceptual ; Low/Design

(c) Low/Conceptual ; Top/Design

(d) Top/Design ; Low/Conceptual

Background

Requirements Determination and Analysis

Elements of Block Diagrams and an Example

Recommendations

Wrap up

Recommendations

1. Determine the active elements (agents).
2. Find the relevant storages and connect them with the agents.
3. Identify which agents communicate with which.
4. Identify the channels between the agents.
5. Check the access arcs and channels: Be more precise about an access (read-only or modifying). Indicate the direction of information flow on a channel. For example, is it a request-response communication? Also, see whether you can add the type of the request the type of data that is transferred.
6. Give each element a proper name. Avoid generic names, such as “computer”, “agent”, “storage”, “memory”, or “data”. The name should reflect what the agent is doing, and what information can be found in a storage.

Guidelines for completing your assessment are available in the DC2 assessment specification file and in the video records of the lecture.

Recommended Reading

- “FMC Quick Introduction” by Andreas Knopfel – available on Blackboard
- “FMC Compositional Structures Block diagrams -Reference Sheet” –available on Blackboard

Background

Requirements Determination and Analysis

Elements of Block Diagrams and an Example

Recommendations

Wrap Up

Drawing and modelling it all together

- Building software-intensive systems is often a complex process, which requires a high degree of efficiency to communicate among stakeholders (anybody who is affected by the project).

- It would hardly be possible to efficiently develop such a system without efficient ways to communicate design ideas among the design team members.

- FMC diagrams facilitate this communication about ideas among analysts, designers, developers, managers and team leader.

Questions?

a.nili@qut.edu.au

Acknowledgment: *some of the learning material are based on the SAP website and blogs (a prominent international consultancy organisation in the area of enterprise systems and system architecture: references have been presented). FMC notations are based on fmc-modelling.org and the Designing for IT unit in semester 2 of 2017.*