

On Architectures, Frameworks, and Models in Thesis Writing for Computer Science

Zaharin Yusoff

Department of Computing and Information Systems, Faculty of Science and Technology,
Sunway University, No. 5, Jalan Universiti, Bandar Sunway, 47500 Selangor, Malaysia
email: zahariny@sunway.edu.my

Abstract – *Thesis writing can be a very complicated, confusing, and exhausting process. This paper looks at the why, how and when in using architectures, frameworks and models for thesis writing in computer science and related domains, as well as achieving clarity in thesis writing in general, with illustrations and examples. The underlying objects in computer science are data and processes, and these are put together into systems and applications. Architectures and frameworks are used to places all the components required to implement systems and applications, while models are used to mimic behaviour, and can then be used to explain, control, and predict. Good expositions of the relevant architectures, frameworks and models in a computer science thesis (or project) would then go a long way in situating the project, emphasising the main contributions, as well as determining what would be left for future work – the basic ingredient for thesis writing (or project reports).*

Keywords: Architectures, component diagrams, frameworks, models, thesis writing.

1 Introduction

Writing theses can indeed be a very complicated, confusing, and exhausting process. Many take a long time to complete, and some even feel like giving up at some stage or another. It is observed that those who manage to finish in good time are those who have a certain level of clarity in their work. Attaining such a level of clarity should be a focus for postgraduates, and they should be closely aided by their supervisors and co-supervisors.

Any work in computer science principally involves data, processes and systems/applications, and these can be placed within architectures, frameworks and models. The proper use of these concepts or objects can go a long way in helping to achieve clarity in one's work, in particular in terms of positioning the work, specifying the objectives and deliverables, and in particular to describe the end results. This is because these underlying objects have very clear structures, and it can also be seen that a similar kind of structuring applied to thesis writing can also help in attaining clarity of its presentation.

This paper will first look at architectures, frameworks and models in general, especially in terms of the why, how and when they are to be used in thesis writing. This will be followed by a look at thesis writing in general, with the roles of architectures, frameworks and models emphasised. The same principles may also be used for writing proposals and reports for research and commercialisation projects.

2 Architectures, Frameworks, and Models

The underlying objects in computer science are data, processes, and systems/applications, which are depicted in Figure 1 below. It is with these three that all work in computer science are carried out – as the ingredients as well as the end products. In as much as data is always tangible, processes and systems/applications may come in the form of software modules or programs, as well as physical devices (typically in Computer Engineering), but they can also be manual actions/activities (typically in Information Systems).

Data begins as Raw Data – unstructured data (e.g. text), structured data (e.g. numbers, tables, vectors), audio, images, video, signals, multimedia, etc. These would be given some form of structure to be kept in some container to form one of the two basic components, generally referred to as Data (the barrel in Figure 1).

The other basic component is Process, which necessarily has an input and an output. The input and output are usually data, with the Process making some form of transformation on the data. There is also another form of output, which is

an action, typically for devices (such as a robot raising an arm, a light being switched on, etc.), or as mentioned earlier, a manual action/activity – these latter two types are general referred to as Procedures.

Data and Processes are then put together to form Systems or Applications (the former being a more general form of the latter), and these are essentially more complex Processes.

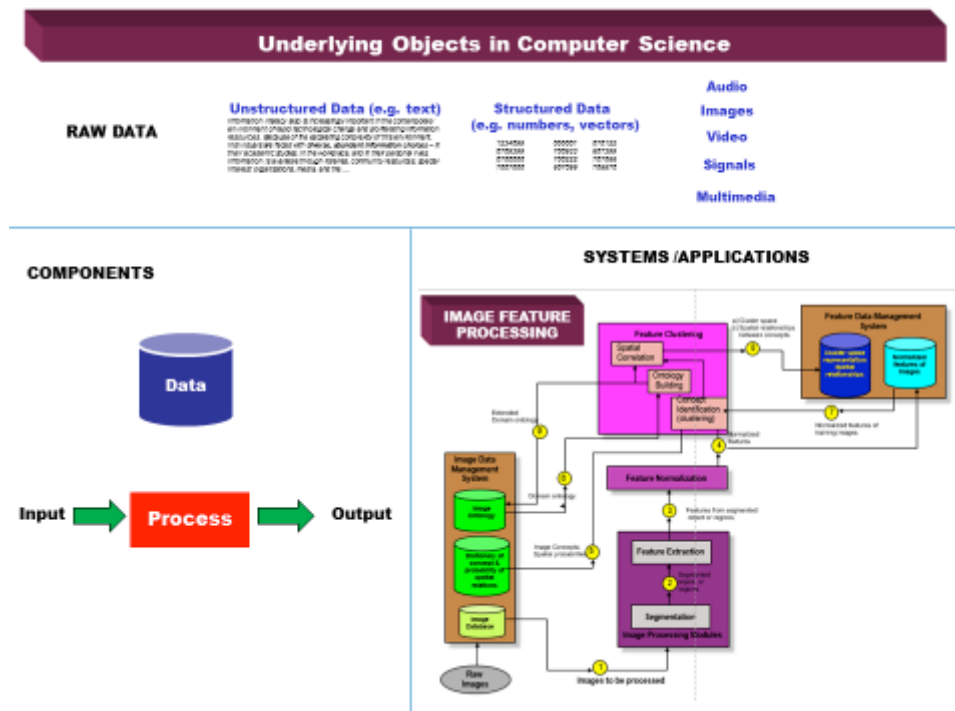


Figure 1: Underlying objects in Computer Science projects

For any documentation of work in computer science (including for a thesis), there is obviously a need to describe the data, processes, and systems/applications involved, especially when they work together. Directly related to these are the concepts of architectures, frameworks and models, which we will now look at. It is important to note that they have different objectives, and they may cover more than the structure of the end result, possibly also the thought processes that are involved in getting the results, and the possibilities of future work.

The way to read the rest of this paper is the following:

- Understand how architectures, frameworks and models should be used (this Section 2).
- See how theses (and reports) should be written in general to attain clarity (Section 3).
- Look at an example of modelling (Section 4).

2.1 Architectures

First, recall that all computer science projects involve data and processes to form systems/applications. This combination may form an **Architecture**. There are many definitions for this, but Business Dictionary (2016) has a very simple yet comprehensive one (<http://www.businessdictionary.com/definition/architecture.html>).

Overall design of a building, structure, or system that unifies its components or elements into a coherent and functional whole.

This is precisely what we have been discussing above, being a unification of the data and processes into a unifying whole. The keyword here is ‘functional’, as the combination is to work together towards a defined goal. Architectures, however, have the connotation of having very precise specifications at every point in the diagram. As such, a looser reference is called a **Component Diagram**, where some of the details may be missing but it still provides a description of a functional whole.

An example of an architecture had been given in Figure 1, which is repeated in Figure 2 below. It is that of an image feature processing module, with images as input and their features as output.

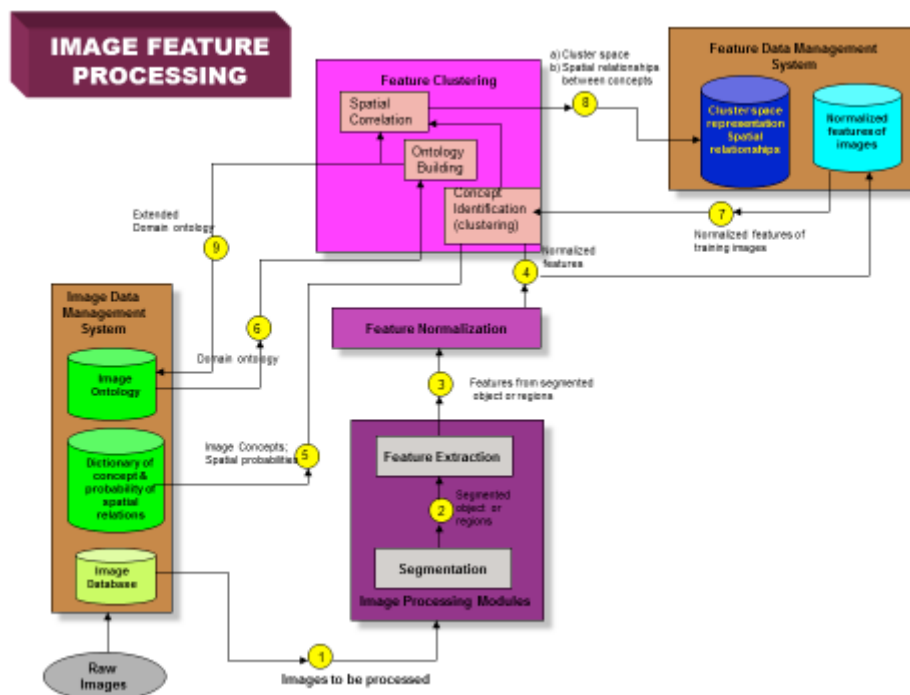


Figure 2: Architecture of an Image Feature Processing module

The architecture places all the components required to implement an image feature processing module, the data as well as the processes. A full scale architecture would have specifications for all the various input and outputs of the processes – namely all the data (barrels) and the small numbered circles. A basic implementation may make use of only a subset of these (e.g. the middle vertical part), but it has to lead from the initial input to the final output.

2.2 Frameworks

An architecture (or component diagram), however, usually refers to a combination of specifically known and well-identified set of data and processes. It is possible that the processes (modules and programs) may be replaced with other equivalent ones, or that some way not be very well-defined yet. Even certain larger subcomponents of the architecture may be substituted with other equivalent sub-combinations to perform the same functions, possibly with better results. A more generic combination of data and processes, where sub-components may be substituted is referred to as a **Framework**. Again, BusinessDictionary (2016) has a very simple yet comprehensive definition (<http://www.businessdictionary.com/definition/framework.html>).

Broad overview, outline, or skeleton of interlinked items which supports a particular approach to a specific objective, and serves as a guide that can be modified as required by adding or deleting items.

An example of a framework is given in Figure 3, which is of a watermarking framework within a biometric face recognition system to enhance security while maintaining the efficiency of the biometric system, (Rizal, 2016). A framework does look very much like an architecture, but it may be missing some of the details. This may well be done on purpose, to allow for flexibility in choosing the most appropriate or available sub-modules (processes) as long as they perform the same specified functions. This framework does not specify any particular Scanner, Feature Extraction, Matcher, Storage, and Application Device to use – for instance, the Feature Extraction module (c) may be any of the modules used in Figure 2. The objective of this framework is to specify exactly what watermarks to use (timestamp and logo), and where they will be tested for.

In general, architectures are used for developing systems/applications, while frameworks are used to add values.

Generic Biometric Authentication System – Watermarking Framework

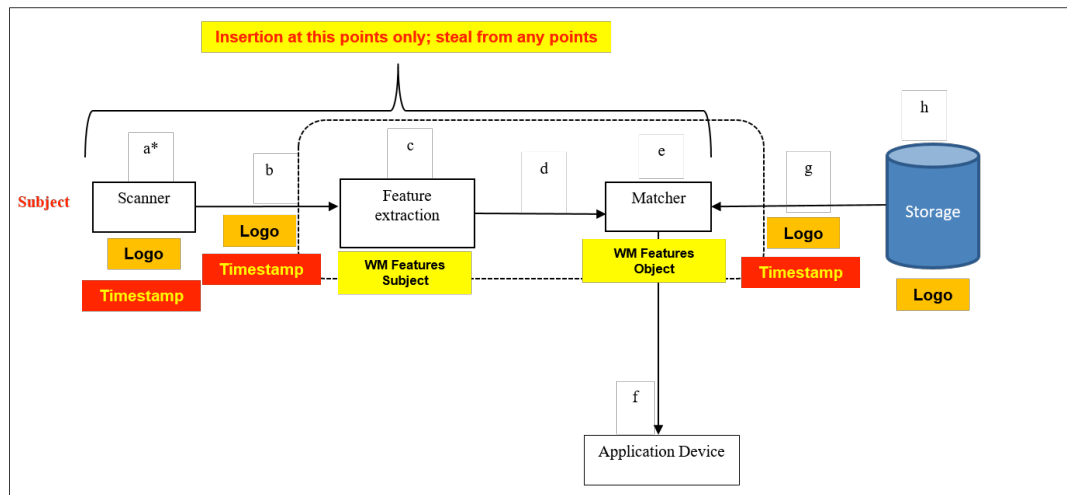


Figure 3: Framework for a Secure Watermarking Technique in a Biometric Face Recognition System

2.3 Models

While Architectures (or Components Diagrams) and Frameworks can be quite readily visualised because they are much more tangible, **Models** are much more complex. Before going into the details of Models, it is perhaps best to mention that all three (architectures, frameworks, models) are at a much higher level than another term also quite often used – Prototype, which is essentially a quick implementation of the three to test out functionalities in view of moving forward. Again, a simple definition is given by BusinessDictionary (2016) (<http://www.businessdictionary.com/definition/prototype.html>).

Pre-production model of a product, engineered for full service test. Changes based on test results are incorporated into the prototype which undergoes the same tests again. On achieving the desired results, the product is approved for volume production.

Having put Prototypes aside, the most complex concept of Models is defined by BusinessDictionary (2016) as follows (<http://www.businessdictionary.com/definition/model.html>) [slightly edited]:

Graphical, mathematical (symbolic), physical, or verbal representation or simplified version of a concept, phenomenon, relationship, structure, system, or an aspect of the real world. The objectives of a model include (1) to facilitate understanding by eliminating unnecessary components, (2) to aid in decision making by simulating 'what if' scenarios, (3) to explain, control, and predict events on the basis of past observations. Since most objects and phenomenon are very complicated (have numerous parts) and much too complex (parts have dense interconnections) to be comprehended in their entirety, a model contains only those features that are of primary importance to the model maker's purpose.

Models range from simple sketches to computer programs with millions of lines of code, but all of them have one thing in common: some elements of the actual 'thing' are abstracted or mapped into the model. Models are divided into three classes on the basis of their degree of abstraction (a) Iconic model: least abstract, physical, 'look-alike' model, such as a model airplane or train. (b) Analogous model: more abstract but having some resemblance to what it represents, such as a chart, graph, map, network diagram. (c) Symbolic model: most-abstract model with no resemblance but only an approximation to what it represents, such as a mathematical equation or formula, financial statement, language, and set of accounts. [See also mental models.]

It is clear that, other than (a), the general definition of a model refers to a rather complex and abstract object. The key terms here are in (1) & (3):

(1) *Facilitate understanding* – which means architectures and frameworks are partially models.

(3) *Explain, control, and predict* – hence a lot of mathematical modelling is carried out on modelling of water and wind currents, epidemics, etc. This is not to be confused with the effort to 'see' how these work, which is in the domain of visualisation techniques.

The point (2) *Decision making*, is basically a usage of models – not an expected feature, and the point (b) *Analogous model*, is essentially a part of visualisation. The point (c) *Symbolic model*, states that points (1) and (3) are very fundamental to models, and as such models need not have to be presented as architectures and frameworks (but they would certainly help).

While architectures, frameworks emphasise components and functionalities, models emphasise behaviours and their predictability. We will look at an example of a model in Section 4, but for the moment we will look at how architectures, frameworks and models can and should be used in thesis writing (and in proposals and reports).

3 Thesis Writing

Writing in general is about presenting the WH points – What, Why, How, Who, When, Where, How Much. Writing blueprints, and to some extent proposals for research grants, would require all of these. Research papers and theses tend to focus mainly on the first three – What, Why and How. Architectures, frameworks and models deal precisely with these, giving the responses via very clear structured diagrams. This is why it is important to use them to attain clarity.

For a thesis, the number of chapters may vary depending on regulations and content, but they all have to cover the essence as given in Figure 4 below.

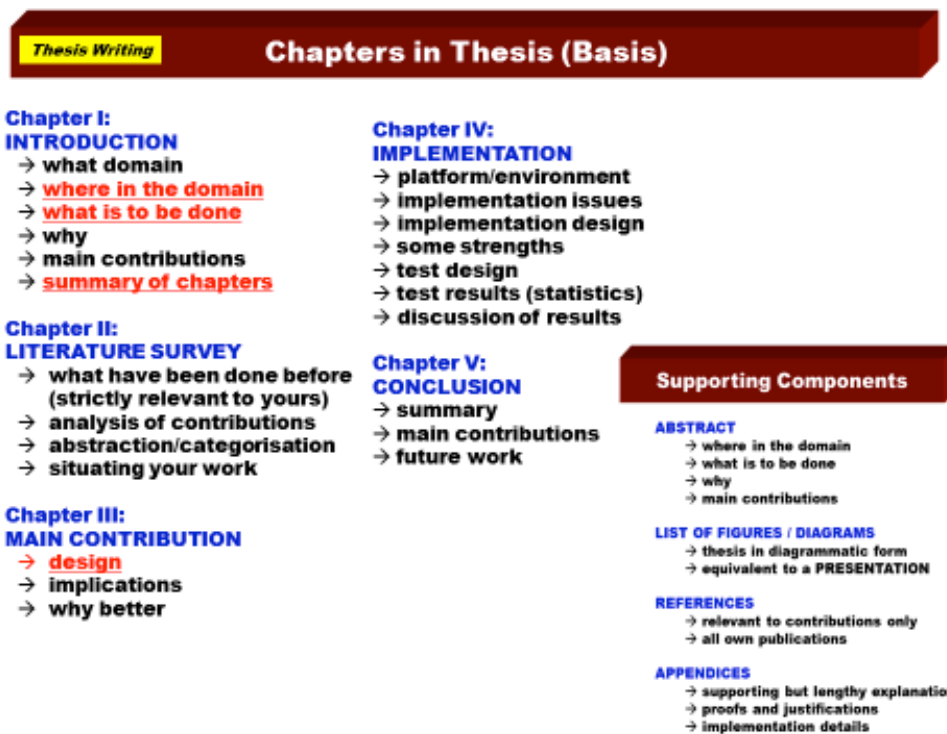


Figure 4: Essence of contents in a thesis

Within the points given in Figure, to attain clarity, it is important to make clear the following right at the outset of the thesis (the same for a research proposals and reports):

- Where in the domain (3.1 below),
- What is to be done (3.2 below),
- Summary of chapters (3.3 below).

The points on why and main contributions are equally important, only that these involve text (and hence not the main emphasis of this paper), and one would also need the the first two to specify the why and the main contributions. The summary of chapters will go a long way in helping the examiner (or evaluator) to read the rest of the document, and it also involves a form of architecture – namely that of the thesis (or report/document).

The reason for the above is that examiners (and evaluators) would usually look for the essence of the thesis (or report), namely the following:

- Clear Deliverables,
- Research value (addition to knowledge),
- Complexity of problem (the challenge),
- The ‘Clever’ parts (e.g. intelligence).

Again, this is the reason why it is best to declare the said three points up front, which can be shown via diagrams for architectures, frameworks and models. Such diagrams will show the clear deliverables within the proper context, from which the research value, complexity of problem and the ‘clever’ parts may be deduced (with support from some text within the presentation).

The diagrams should then be supported by further details (diagrams) in:

- Design, in the chapter on Main Contribution (3.4 below).

All in all, a good/clear thesis (or report) should be readable solely via the diagrams. The text should be written around the diagrams, and a good test is that when all the text is deleted, the remaining diagrams would serve as the presentation slides for the viva (or project) once the title, thank you, and other minor (cosmetic) slides are added. The diagrams in this paper is a demonstration of this point.

3.1 Where in the Domain

Situating the research is important to avoid confusion and any not very relevant questions that may arise. This is where a diagram giving the architecture/framework of the whole domain (domain conceptualisation) would be very useful, where situating the research would simply be pointing to some position in the diagram. As an example, Figure 2 would serve very well to situate a research – such as a thesis (or project proposal/report) that works on the development of an ontology on images (left hand side of Figure 2).

The diagram further below, Figure 5, gives an example for the domain conceptualisation for Bioinformatics – where the left hand side shows the more explored areas of research, while the right hand side shows the more unknown territory. Many diagrams for such a purpose can be obtained in various R&D Roadmaps (and their references), such as (MIMOS, 2007), (UNITEN, 2007), (MMU, 2008), (UNIMAS, 2013), etc.

3.2 What is to be Done

This is where diagrams for architectures, frameworks and models are best used – and they should be declared right at the beginning. The grain-size of the components is important, for example:

- Figure 2 is quite all right, by pointing to the component targeted (e.g. ontology of images), but preferably with a forward reference to a more detailed architecture under Design in Chapter 3 (see within Figure 4).
- Figure 3 is fine, as it clearly shows what watermarks are proposed and where.
- Figure 5 is rather general, serving its purpose for situating the research, but another diagram for the proposed architecture, framework or model should be provided for the chosen component.
- Figure 6 below gives a good example, which is a framework that shows the entirety of what is to be developed (Al-Shrouf, 2006). Figure 6 essentially states that to develop the ultimate goal on the right hand side, those on the left will have to be developed first and in the given sequence and combinations. There are of course other diagrams in Chapter 3 (under Design) that would detail out each of the components above.

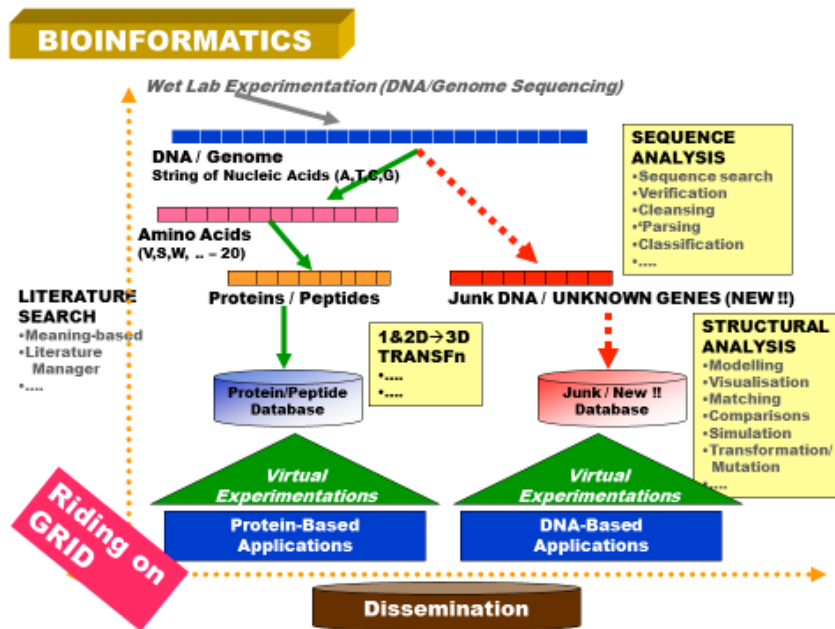


Figure 5: Domain Conceptualisation for Bioinformatics

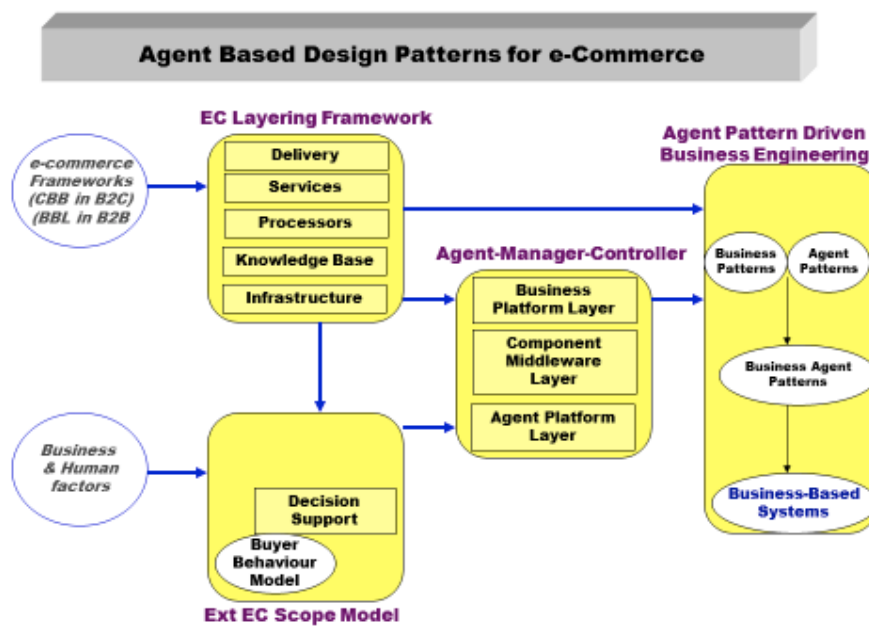


Figure 6: Example of a Framework showing the entirety of a project (thesis)

3.3 Summary of Chapters

As mentioned earlier, a summary of chapters will go a long way in helping the examiner (or evaluator) to read the rest of the document, and it also involves a form of architecture – namely that of the thesis (or report) document). By having such a diagram, the examiner (or evaluator) will be guided as to how the thesis (or report) is to be read. It would also show the development of the arguments or the logic/theme of the whole project, all within a single diagram. Most of all, it also helps the author map his/her thought processes down, and consequently the writing.

Figure 7 below gives an example (Al-Shrouf, 2006), which helped the candidate to complete writing the entire thesis within 2 months. Interestingly enough, Figure 7 together with Figure 6 given in the first chapter of the thesis were instrumental in the examination of the said thesis, where both external examiners submitted their reports within 3 days of receiving them – as mentioned in their reports and verbalised in the viva.

Figure 7 declares that the work is a result from 3 domains – Business, Information Systems and Computer Science. It shows which subdomains of each of these would have to be covered to get a good understanding of the contributions (Chapter 2: Literature Survey), the main contributions in Chapters 3 and 4 as declared in Figure 6, and a brief mention of Implementation (Chapter 5) and Conclusion & Future Work (Chapter 6).

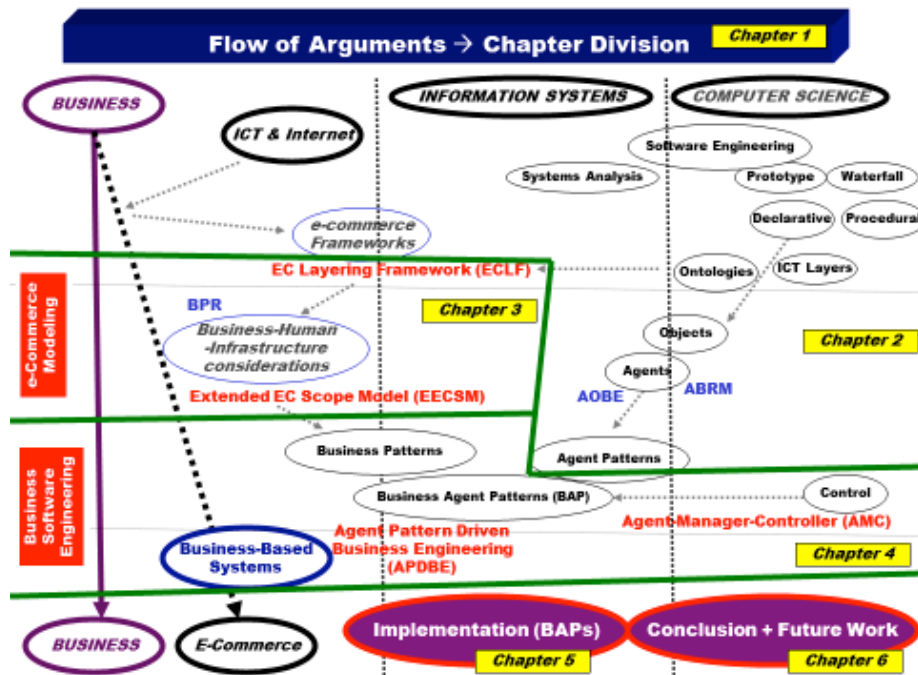


Figure 7: Example of a Chapter by Chapter Summary (PhD thesis)

3.4 Design (Within Chapter on Main Contribution)

The other key diagrams are in Chapter 3 for the main contribution(s), which should give further details of the modules or data pointed out within the earlier diagrams on architectures, frameworks and models. These would be the diagrams to clinch the thesis award (or project proposal/closure), giving a very quick understanding of the contribution, its complexity and 'cleverness'. An example is given in Figure 8 below, for the basis of an algorithm to take over control of an unmanned aerial vehicle (UAV) to obtain the exact coordinates of a victim in a search and rescue (SAR) mission at sea, (Nor Wahidah, 2015). This is only for illustrative purposes.

This completes the discussion on thesis writing. The chapters on Literature Survey, Implementation, and Conclusion & Future Work are also important, only that they would be much more readily and better understood if the architectures, frameworks and models are well-presented and explained.

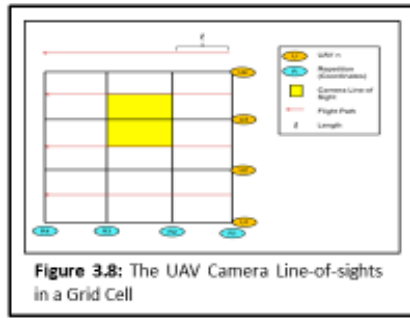


Figure 3.8: The UAV Camera Line-of-sights in a Grid Cell

NOTE 2

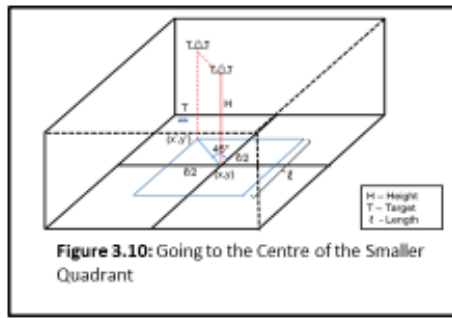


Figure 3.10: Going to the Centre of the Smaller Quadrant

NOTE 3

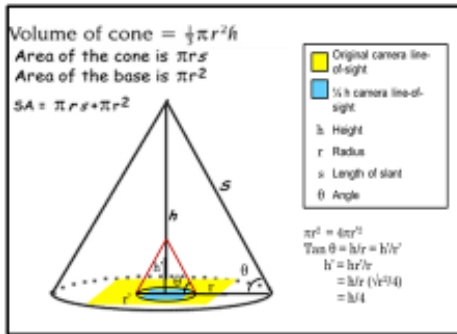


Figure 3.11: Descending to a Lower Height to Fit the Camera Line-of-sight exactly to a New Smaller Quadrant (1/4 of the Original Camera Line-of-sight)

NOTE 4

From Chapter 3

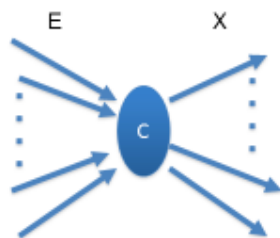


Figure 8: Example of a diagram on Design

4 Modelling : an Example

This last section is only to give an example of a typical model, which is different from an architecture or framework, but is also in diagrammatic form supported by the accompanying definitions. In general, most models are mathematically based, or at least has some mathematical grounding. In computer science, some earlier models include the formal language theories and their equivalent automata (linear languages and finite state machines, context free language and pushdown automata, etc.), grammar formalisms and their operations, and a number of others.

The main point here is that, while architectures and frameworks are clearly structured and may have well-defined specifications, they would still require some mathematical grounding to be considered a model. Again only for illustrative purposes, Figure 9 (a) & (b) give an example for a basis for a model for traffic systems, based on Basic Traffic Units that can be combined to form Traffic Systems, upon which operations research and simulation models may apply in an effort to explain and predict congestions, (Khairani, 2016).



- A **Basic Traffic Unit** is a triple (C,E,X):
- C is the central point where all roads in the unit meet, which can be a –
 - Junction flowing in one direction
 - Toll plaza, in which case (C,E,X) is a **Toll Plaza Area**
 - E is a set of roads entering C, with the number of roads |E|
 - X is a set of roads exiting C, with the number of roads |X|

From the definitions above, C is a:

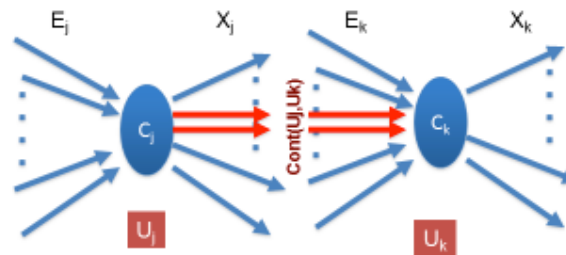
- **Bottleneck** if $|E| > |X|$
- **Funnel** if $|E| < |X|$
- **Straight** if $|E| = |X|$

Figure 9 (a): Example of a Model – Basic Traffic Unit

A **Traffic System** is a set of interconnecting Basic Traffic Units, i.e. a set of triples $U_i = (C_i, E_i, X_i)$ with $i = 1, \dots, n$.

$U_j = (C_j, E_j, X_j)$ is adjacent to and **precedes** $U_k = (C_k, E_k, X_k)$ if $\text{Intersection}(X_j, E_k)$ is not empty

If U_j precedes U_k , the meeting points of the exit roads X_j and the entry roads E_k will be referred to as the **continuing point** $\text{Cont}(U_j, U_k)$



$\text{Cont}(U_j, U_k)$ can also be a bottleneck if $|X_j| > |E_k|$ (see the example in the next slide), which will be referred to as a **derived bottleneck**

Figure 9 (b): Example of a Model –Traffic Systems

5 Conclusions

This paper has one main objective – to look at how theses (as well as proposals and reports) can be written with the proper use of architectures, frameworks and models in order to attain the necessary level of clarity to speed up thesis writing as well as present the contents in such a way to facilitate easier understanding. The use of these objects have been covered in terms of the why, when and how, emphasising on their absolute necessity. Proper thesis writing has also been covered in view of attaining clarity, with recommendations on the underlying points to be covered, in particular via a certain set of diagrams. This paper is not just about modelling techniques, but also about modelling thesis writing. The same principles may also be used for writing proposals and reports for research and commercialisation projects.

References

- Al-Shrouf, F.M. (2006). *Agent-based design patterns approach and its applications for business-to-consumer e-commerce*. PhD Thesis, Universiti Sains Malaysia, Penang, Malaysia.
- BusinessDictionary (2016). <http://www.businessdictionary.com/definition/>
- Khairani, A.M. (2016). *Queue Analysis at Toll Plazas (Inbound): A Basic Model for Traffic Systems Towards a Study of the Effect on Toll Plazas With Different Arrival Patterns*. PhD Thesis (in submission), Universiti Kebangsaan Malaysia, Bangi, Malaysia.
- MIMOS (2007). *Applied Research Commitment (ARC): Knowledge Technology*, Knowledge Technology Laboratory, MIMOS, Kuala Lumpur, Malaysia. (83p).
- MMU, (2008), *MMU R&D ROADMAP: 2008-2013 (for 3 High Impact & Niche Areas, and includes 4 Flagship Projects)*, Multimedia University, Cyberjaya, Malaysia. (1,024p)
- Nor Wahidah Z. (2015) *Establishing a Grid Pattern Model for Ground Search and Rescue (SAR) Area*, MSc thesis, Universiti Pertahanan Nasional Malaysia, Kuala Lumpur, Malaysia.
- Rizal, I. (2016). *Watermarked Face Recognition Scheme – Enhancing Security While Maintaining Effectiveness of Biometric Authentication Schemes*. PhD Thesis, University of Portsmouth, England.
- UNIMAS (2013), *Universiti Malaysia Sarawak: FCSIT R&D ROADMAP 2012–2020 (for High Impact & Niche Areas, and including Flagship Applications)*, Kota Semarahan, Sarawak. (327p)
- UNITEN (2007). *UNITEN R&D ROADMAP: 2008-2015 (for 4 High Impact & Niche Areas, and includes 5 Flagship Projects)*, Universiti Tenaga Nasional, Kajang, Malaysia. (1,004p).