A DYNAMIC ITINERARY

Getting from A to B

ANU
ENGN2225

Table of Contents

1. Introduction & Problem Definition

For the ENGN2225 72 Hour Take Home Exam students were given a choice of two problems to tackle. These were devising a temperature control solution for a university student's house, and efficiently getting the executives of a Canberra based technology business to fortnightly meetings in Sydney. Here, the latter option was chosen on account of its challenging nature, openness to innovative solutions, and wider potential for application to real world scenarios. With the use of the systems engineering process, a wide range of conceptual solutions were analysed and the most suitable concepts were identified.

The problem definition states that the client is a Canberra based small business that specialises in providing energy saving technological solutions to corporate offices. They have a contract with a business associate in the Sydney CBD which requires the executives to attend fortnightly face-to-face meetings at the Sydney office. The client's Chief Executive Officer (CEO) must attend all meetings with one of either the Chief Technical Officer (CTO), Chief Financial Officer (CFO) or the Operations Manager (OM). Sometimes all of these people are required to attend. These meetings are usually in the late morning and last for about an hour and a half on average.

The key wants and needs of the client were that the solution had to be practical and time efficient. They also desired that the solution be of minimal cost but were more concerned that it would have to be reliable and convenient. They have also outlined that they are open to implementing more than one solution, if this option seems best. Last but not least, the client wants to implement a solution that supports their philosophy as an energy-saving business.

The solution required here only needs to be implemented for a year, after which video-conferences can replace face-to-face meetings when necessary. However, adaptability in the solution that would allow its use to continue indefinitely would certainly also be a desirable trait in the case.

2. Requirements Engineering [1] [2]

With the wants and needs of the client, outlined in the Problem Definition above, a list of explicit Customer Requirements was compiled as the first step of the Requirements Engineering process. Six requirements were identified; Practical, Time Efficient, Affordable, Reliable, Convenient and Appropriate. These requirements were then paired against each other to determine their relative importance in a Pairwise Analysis matrix (Table 1).

Table 1: Pairwise Analysis of Customer Requirements

	CR-01	CR-02	CR-03	CR-04	CR-05	CR-06	Sum	Rank
CR-01 Practical	-	1	1	1	1	1	5	**1st**
CR-02 Time Efficient	0	-	1	1	1	0	3	**3rd**
CR-03 Affordable	0	0	-	0	0	0	0	**6th**
CR-04 Reliable	0	0	1	-	1	0	2	**4th**
CR-05 Convenient	0	0	1	0	-	0	1	**5th**
CR-06 Appropriate	0	1	1	1	1	-	4	**2nd**

According the Pairwise Analysis matrix in Table 1, the most important requirement is practicality. The solution needs to get the executives to the meeting no matter what, or else the client will risk breaking their contract. The second most important requirement is the appropriateness of the solution to the business philosophy followed by time efficiency. It is important to stay true to your cause even if this means spending a little extra time (and money). Reliability, convenience and affordability come fourth fifth and sixth respectively, as outlined by the client's preferences in the problem definition. Each of these requirements were then decomposed into Technical Performance measures (Table 2).

Table 2: Technical Performance Measures for Getting from A to B

Customer Requirement	Design Attribute	Engineering Characteristics
Practical	Functional	Gets from CBR to SYD (Y/N)
	Realistic	Uses Existing Technology (Y/N)
Appropriate	Energy Efficient	Joules per Person ($\log_{10}$ J)
Time Efficient	Minimised Downtime	Total Journey Time (Hours)
		Productive Time (Hours)
Reliable	Self-Reliant	Third Party Reliance (Y/N)
Convenient	Simple to Plan	Number of Organising Steps (#)
Affordable	Minimised Cost	Cost per Trip per Person ($)

The list of Technical Performance Measures was kept short for the sake of simplifying the analysis. There are a lot of Yes/No conditions in the Engineering Characteristics section, largely because of the nature of the problem where it is literally as simple as getting from A to B. There is not too much concern with the details of the operation of each of the concepts, just how well they can address the problem.

It should also be noted that it was decided to measure energy consumption with a logarithmic scale for comparison, since many solution concepts will have wildly different magnitudes of energy use. Following this, a small house of quality was made to explore the relationships between the Design Attributes and Engineering Characteristics (Figure 1).

		+9	+3	+9					
			+9		+1				
						+1			
					+3		+3		
		+9		+9	+3	+1	+9	+3	
Key + Positive Relationship - Negative Relationship = Optimal Value 1 Weak Relationship 3 Medium Relationship 9 Strong Relationship	Weighting	+Gets from CBR to SYD	+Uses Existing Technology	-Joules Per Person	-Total Journey Time	+Productive Time	=Third Party Reliance	-Number of Organising Steps	-Cost per Trip per Person
Functional	6	9	3		9		3		
Realistic	6		9	3	1		9		3
Energy Efficient	5		1	9	9	3			1
Minimised Downtime	4				9	9			
Self-Reliant	3		9				9	3	
Simple to Plan	2		1					9	3
Minimised Cost	1		3	3	1	1	3	3	9
Metrics		y/n	y/n	J	hrs	hrs	y/n	#	$

Figure 1: House of Quality (HoQ)

Judging from the House of Quality in Figure 1 most of the Design Attributes are relatively independent from the Engineering Characteristics since there are no more than 2 strong (9) relationships in the body of the HoQ. This is reinforced in the roof of the HoQ where there are not many strong relationships between the Engineering Characteristics. The most prevalent relationship is whether a third party is involved or not, which seems to affect costs and time efficiency overall.

The conclusion that can be drawn from this Requirements Engineering analysis is that largely the requirements for this problem are relatively independent from each other. Subsequently, this would not make for a very interesting or meaningful further analysis of simple concepts, for example driving a car vs taking a plane. Thus, it was decided at this stage, that a modification to the conventional systems approach would be taken.

Instead of only comparing transport systems, extra classes of supplementary systems could be added to make the solutions more vivid and comprehensive. The independence of the Technical Performance Measures can be utilised to allow the additional classes of systems to be mixed and matched at will, forming subsystems of the overall solution. These extra system classes should be analysed separately, to the same criteria, so that the best systems in each class can be picked out where it suits the client to create solution(s) best tailored to the problem.

3. System Function Definition [1] [3]

Expanding on the idea of system classes, different kinds of systems could be used to achieve different Customer Requirements. Together a combination of systems from all the different classes would be able to satisfy the client's requests. Because of the independence shown in the HoQ (Figure 1) there would be minimal clashing effects between systems from different classes during the implementation. Hence, at the concept generation stage, the first step was to figure out how many classes would be needed.

Obviously, a Transport class needs to be considered, which addresses the Practical aspect of actually getting the executives from Canberra to Sydney and back. In order to be Time Efficient and subsequently get value for the client's money, the productivity of the trip must be increased. Evidently it is incredibly inefficient to drive 6 hours for a 90 minute meeting. This emphasises the need for a Productivity class which will provide additional business for the executives to attend to, killing multiple birds with one stone. Lastly, because fortnightly trips to Sydney is counter intuitive to the business' cause, there needs to be a system that counteracts that. This class is the Offset class.

Instead of making Concept Classification Trees to generate concepts that satisfied the individual Customer Requirements, 'Class Classification Trees' were made to suit the purposes of each of the three classes. These 'trees' are listed in Tables 3, 4 and 5 for convenience, as the systems within each of the classes can all be seen at once. The systems within each class that were deemed appropriate and taken to the next level of analysis are highlighted in **bold**.

Table 3: Class Classification Tree for Transport

Transport		
Self-Reliant	Third Party	Alternatives
Personal Car	**Plane**	Hitchhike
Motorbike	**Bus**	Horse and Cart
Walk	**Train**	Private Jet
Cycling		Live in Sydney
Skating		

Table 4: Class Classification Tree for Productivity

Productivity		
Financial	Technical	Operational
Investments	**Product Inspections**	**Attend Meeting***
Financing	**New Installations**	**Replenish Inventory**
Accounting	**Research & Development**	**Visit Consultants**

Table 5: Class Classification Tree for Offset

Offset		
Publicity	Monetary	Networking
Attend Conferences/Exhibitions	**Overnight Stays**	**Community Outreach**
Marketing	Seek Sponsorship	Green Programs
Propaganda		Pro Bono Work

Every considered solution to the problem is a combination of at least one of the systems within each class. Choosing a particular system in a class does not really rule out options from within other classes because of their independence. By using this combinational method, thousands of different trip itineraries can be devised. It is certainly impractical to explore the operation of each and every permutation and combination of systems so a very generic concept flow chart has been devised to explain how the systems fit together. This flow chart is very reminiscent of a Functional Flow Block Diagram except the boxes can be dynamically replaced to reflect which kind of class or system is being considered, this is done by implementing if statements into the diagram, as shown in Figure 2.

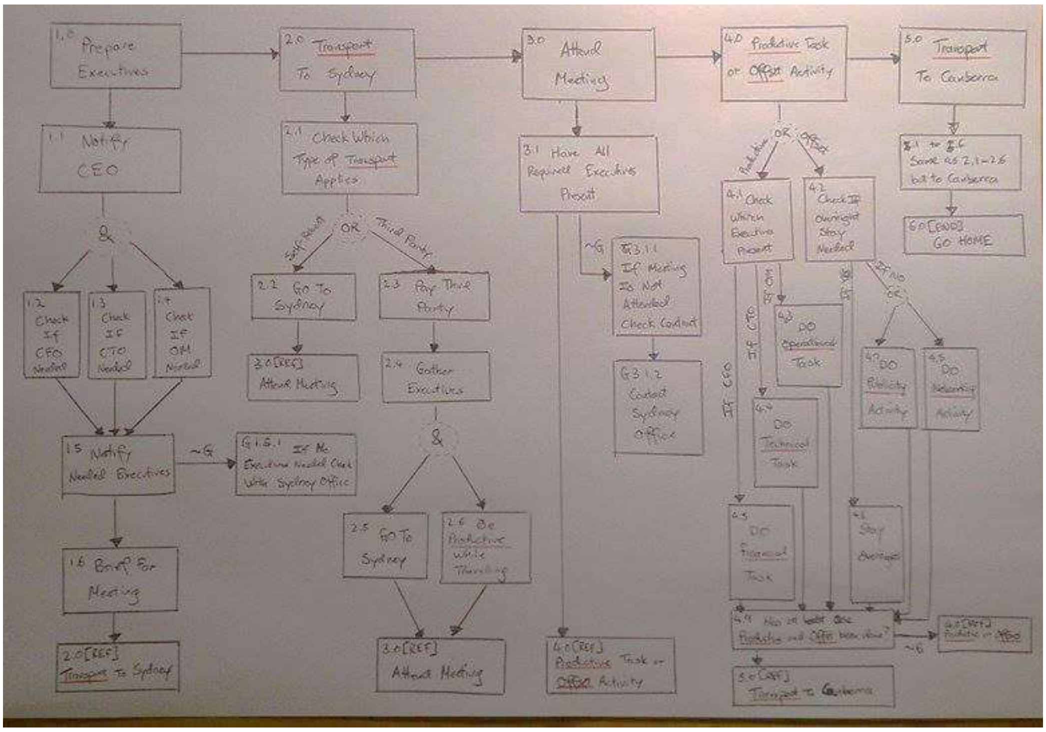


Figure 2: Functional Flow Block Diagram/Design Concept (Also Appendix 1)

From the diagram in Figure 2, it is possible to see that the flow block diagram, which details the use of the solution, is very much the solution itself since ultimately the individual systems combine to form a schedule of events for the client to follow. Rather fittingly to the theme of the problem, the diagram in Figure 2 serves two purposes at once, describing the solution and its operation simultaneously. From here on in, this concept will be referred to as the ‘Dynamic Itinerary’.

4. Subsystem Integration [1] [4]

Great care has been taken to not mention the term 'subsystem' interchangeably with 'class' up until this point in the documentation. Whilst the system classes appear to be analogous to subsystems in this situation it was decided to keep the two terms distinct. This was done to reflect the fact that system classes are formed from a list of distinct concepts whereas subsystems are generated by decomposing a concept into smaller parts. However they are similar in the aspect that both classes and subsystems combine and interact amongst each other together to form the overall solution.

Because flexibility and adaptability is favourable in the design, it is a much better approach to analyse interchangeable concepts as opposed to specific workings of static solution designs. Hence the decision was made to apply the Subsystem Integration process on the classes as a whole instead of concepts within each of the classes. Not only does this stop the scope of the solution from narrowing, but also saves a significant amount of time. Considering the Dynamic Itinerary solution a System Boundary Chart (Table 6) was constructed to identify the internal, external and excluded parts.

Table 6: System Boundary Chart for the Dynamic Itinerary

	Internal	**External**	**Excluded**
Concept: **Dynamic Itinerary**	CEO CFO CTO OM Transport Method(s) Productive Tasks Offsetting Activities	Business Associates Customers General Populace Canberra Sydney Company Money	Meeting Venue Meeting Date and Time Terms of the Contract

The internal systems of the solution were then classified under their relevant system classes (Figure 3); either Transport, Productivity or Offset. Then, the classes were organised into a Functional Block Diagram (Figure 4) to show the interactions internally between classes and with the external stimuli. It is important to notice what variables have been excluded from the scope of this problem. Table 6 shows that the Dynamic Itinerary should work irrespective of the exact meeting venue in Sydney or the date of the meeting. Also the terms of the client's contract with their business associate (and what happens if they break the contract) are beyond this solution's concern.

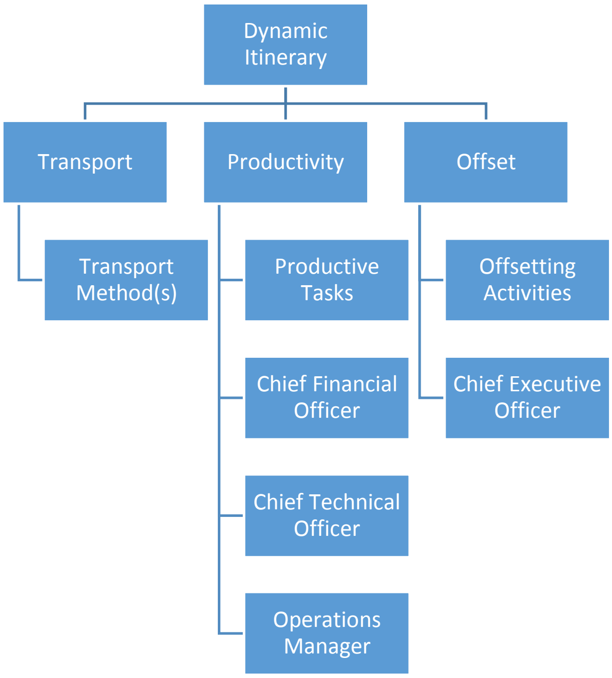


Figure 3: Functional Allocation for the Dynamic Itinerary

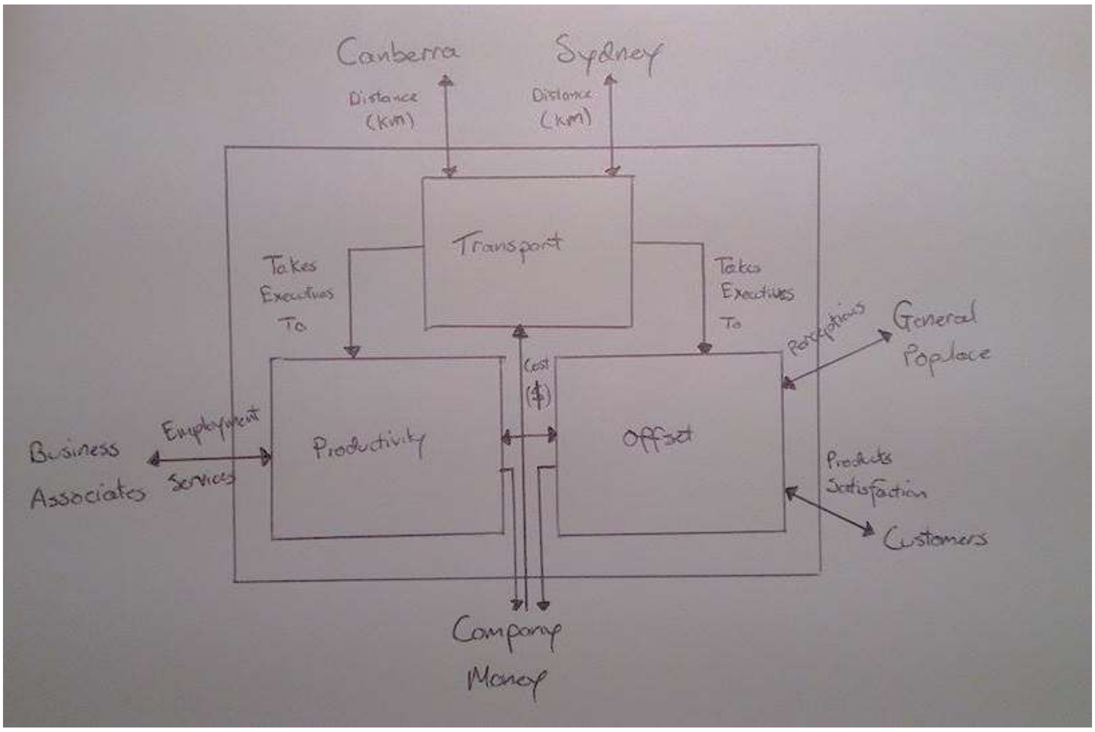


Figure 4: Functional Block Diagram for the Dynamic Itinerary

Here the interactions between the classes within the system boundary are evidently scarce, once again reinforcing their independence. The external influences are also listed showing, rather qualitatively, how the concept classes react to their respective external stimuli. It is interesting to note that the most important part is Company Money, which has an effect on all of the classes because in the end businesses cannot survive without money.

5. System Attributes [1] [5]

A System Attributes cascade (Table 7) was developed to help explore the specifics of how to achieve the customer requirements. Furthermore, it helped to explicitly link the classes with respective attributes. Once again this technical specification documentation was not expanded too deeply for the sake of simplifying analysis.

Table 7: System Attributes Cascade

Customer requirement	**Primary Attribute**	**Secondary Attribute**	**Tertiary Attribute**	**Classes**
Practical	Functional	A1.1 Gets from Canberra to Sydney and back	A1.1.1 Motorised	TRAN
			A1.1.2 Fast	TRAN
	Realistic	A2.1 Uses existing technologies	A2.1.1 On the market solution	TRAN
Appropriate	Energy Efficient	A3.1 Low energy consumption per capita	A3.1.1 Minimised wasted energy	OFF
			A3.1.2 Justified energy use	OFF
Time Efficient	Minimised Downtime	A4.1 High proportion of productive time spent	A4.1.1 High Productive Hours	PRO
			A4.1.2 Reasonable Total Journey Time	TRAN/PRO/OFF
Reliable	Self-Reliant	A5.1 Minimised reliance on third parties	A5.1.1 Reputable third parties	TRAN
Convenient	Simple to Plan	A6.1 Consistent organisational procedure	A6.1.1 Few number of steps	PRO
			A6.1.2 Similar planning process for all trips	PRO
Affordable	Minimised Cost	A7.1 Low cost per trip per person	A7.1.1 Minimised travel cost	TRAN/PRO/OFF
			A7.1.2 Minimised opportunity cost	TRAN/PRO/OFF

Note: TRAN = Transport Class, PRO = Productivity Class, OFF = Offset Class.

The system attributes cascade goes to reiterate how the classes were designed to fit specific customer requirements. Transport largely concerns practicality and reliability, Productivity affects efficiency and convenience and Offset helps make the solution appropriate to the business philosophy. As mentioned from the Functional Block Diagram (Figure 4), money affects all classes as expected. Total time also depends on what systems are chosen from within each class.

With this knowledge, it was then possible to begin refining the concept of the Dynamic Itinerary in order to figure out which transport methods were favourable and which productivity tasks and offset activities were of greater impact and priority for the trips.

6. Verification & Evaluation [1] [6]

To verify and test whether a concept met the customer requirements, a number of tests had to be designed. The virtue of the problem, Getting from A to B, ended up leaving many of the attributes in the System Attributes Cascade able to be tested simply by inspection. For example, even a fool can determine whether a transport method is motorised and fast or not. Other attributes like reliable third parties and number of steps can easily be researched or simply counted. These inspection-tested attributes are highlighted in green in Table 7.

More complex attributes to verify were ones that required some basic knowledge of how some of the systems within the classes worked. For example minimised travel and opportunity cost will require a little bit of knowledge about accounting and management to prove they are viable. Nevertheless such tests would still be trivial to the professional executive clients and could be carried out by someone like the Chief Financial Officer. These proof-of-concept tests are highlighted in yellow in Table 7.

The two most difficult attributes to test are for Justified Energy Use and Reasonable Total Journey Time. These are both extremely subjective variables and cannot be reliably predicted by theoretical analysis. The best course of action is for the executives to just go on some preliminary trips and get a feel for what seems like reasonable journey time and justification of energy use. This is in a sense prototyping, or as close as the Dynamic Itinerary can ever get to being prototyped without actually being fully implemented itself. These attributes are highlighted in red in Table 7.

All of the systems in each of the classes were then evaluated against the customer requirements to ultimately score them relative to each other. This was done with the use of Weighted Evaluation Matrices. A multitude of these matrices had to be filled out, one of the pitfalls of the modified systems approach taken here, but it was all done to facilitate the implementation of an adaptable dynamic concept. The first concept selection matrix for Transport methods is shown below in Table 8.

Table 8: Weighted Evaluation Matrix for Transport Methods

Design Requirement	**Weighting**		**Personal Car**		**Plane**		**Bus**		**Train**	
	Rank	Weight	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value
Practical	1	6	3	**18**	5	**30**	3	**18**	3	**18**
Appropriate	2	5	0	**0**	0	**0**	1	**5**	3	**15**
Time Efficient	3	4	1	**4**	3	**12**	1	**4**	1	**4**
Reliable	4	3	3	**9**	1	**3**	1	**3**	1	**3**
Convenient	5	2	5	**10**	1	**2**	3	**6**	1	**2**
Affordable	6	1	3	**3**	0	**0**	5	**5**	5	**5**
Totals				**44**		**47**		**41**		**47**
Scale	**5** – *Exceeds Compliance*		**3** – *Full compliance*	**1** – *Partial compliance*	**0** – *Non compliance*					

From Table 8 is clear that there is a strong competition between various transport methods. All of the transport systems would still be in contention for use, yet some more than others. For instance, the plane and train have scored equal highest but their departures are infrequent, favouring the convenience of a car or bus if schedule times do not suit the meeting. The concepts are so closely matched anyway that it really should depend on the specific situation, as well as the client's preferences, to decide which method of transport to take.

The rest of the concepts within the classes were analysed in weighted evaluation matrices which can be found in Appendix 2. Their scores have been listed below in Table 9 for reference.

Table 9: Weighted Evaluation Matrix Scores for All Classes

Transport	Productivity		Offset
Personal Car (44)	Investments (13)	New Installations (46)	Attend Conference/Exhibition (60)
Plane (47)	Financing (17)	Research & Development (39)	Marketing (9)
Bus (41)	Accounting (15)	Replenish Inventory (32)	Overnight Stays (17)
Train (47)	Product Inspection (26)	Visit Consultants (15)	Community Outreach (60)

The scores in Table 9 tell the client in an intuitive manner which activities will be most productive and useful to do whilst in Sydney after attending the meeting. Observing the weighted evaluation matrices for the Transport, Productivity and Offset concepts there is clear failure in some of the concept's addressing of the design requirements. Normally, with a static solution, this would be an instant fail for the concept to progress. However because each of the systems within a class specialises in addressing specific Customer Requirements and not others, shown in Table 7, this is acceptable because together these pieces will come together to create a flexible and effective solution.

Interpreting the scores in Table 9, a highly time efficient itinerary would be for the client to first take a plane to Sydney and attend the meeting in the early morning. Following the meeting they could make some new installations of their product in the Sydney CBD area, if applicable. If not, then they could seek to do some research and development on new products instead or replenish their inventory. Following this they could attend a conference, relevant to the business' interests, if there is one. Otherwise they could reach out to the community, like at schools or workplaces, and get involved with helping everyday people become more energy efficient in their lives. They could finally wrap up the day by taking the last train home to Canberra, or the bus if timetables happen not to suit.

That was only an example of what the client could achieve with this new Dynamic Itinerary system. By following the steps in the flow diagram (Figure 2), the executives can make their own informed decisions on what to do. Obviously, the solution is open to refinement, especially as the testing method used to rank the concepts in Table 8 and Appendix 2 was almost entirely by inspection. Regardless, this evaluation of the Dynamic Itinerary has shown that the system it has its merits.

7. Design Communication [1] [7]

The Dynamic Itinerary system is essentially an optimisation algorithm for designing business trips. Given that the client is a technology company, it would be reasonable to believe that the executives would be able to understand the solution's operation rather easily. However, because the solution is so broad and adaptable, it is entirely possible to repurpose the Dynamic Itinerary to any client.

In this situation, it is not guaranteed that the executives will understand how the system works or how to use it. An innovative course of action would be to write a computer program to implement the algorithm, allowing the user to input their own Customer Requirements and state the relative importance when needed. Better still the program could be turned into a smartphone app. Apps have wide coverage in the business world and are designed to be user friendly and convenient. An app for businessmen to optimise their time and money on business trips would undoubtedly have a market.

To test this concept, a rapid prototype was made using hyperlinks in Microsoft PowerPoint. The prototype gives an insight into what the look and feel of a Dynamic Itinerary app might be. Although it lacks all the content and adaptability that the program would ultimately have, the example user interface shows how the ideas and thoughts of the user can flow into the screen.

8. Conclusion [1]

The systems engineering approach used here to devise the Dynamic Itinerary concept has outlined the operation of the solution to the problem 'Getting from A to B'. With the use of the Dynamic Itinerary system for planning business trips, the client's executives can make the most out of their mandatory fortnightly meetings with their business associates in Sydney. The solution will adapt for every trip, suggesting different activities to be productive and offset the impact of the costs of the trip, depending on the current situation. Even after the client's contract has stopped mandating fortnightly meetings in Sydney, the executives can still use the Dynamic Itinerary to plan any trips of theirs in the future. This solution is so flexible that it can work to satisfy all the customer requirements, given enough user input and feedback.

There is always room for improvement though, the sky's the limit. Potential lies in having the solution interface with the cloud and personal calendars to plan around external events. Alternatively, the internal list of systems within classes could be updated in real time to reflect tasks that need urgent attention or increase the priority of activities that have been neglected for a long time. There are endless ways that the Dynamic Itinerary could be improved and catered to the booming market for productivity applications in the business world. Overall, there is latent power in the design of the Dynamic Itinerary documented here in this report. With some robust improvements, it could very well be the next big app to take the corporate world by storm.

9. Bibliography

[1] ANU ENGN2225 Group T14-B, “Green @ Heart, ENGN2225 Final Group Project Report,” Canberra, 2014.

[2] T. Zhang, “Requirements Engineering Individual Research Paper,” Canberra, 2014.

[3] A. H. Babaei, “System Function Definiton Individual Research Paper,” Canberra, 2014.

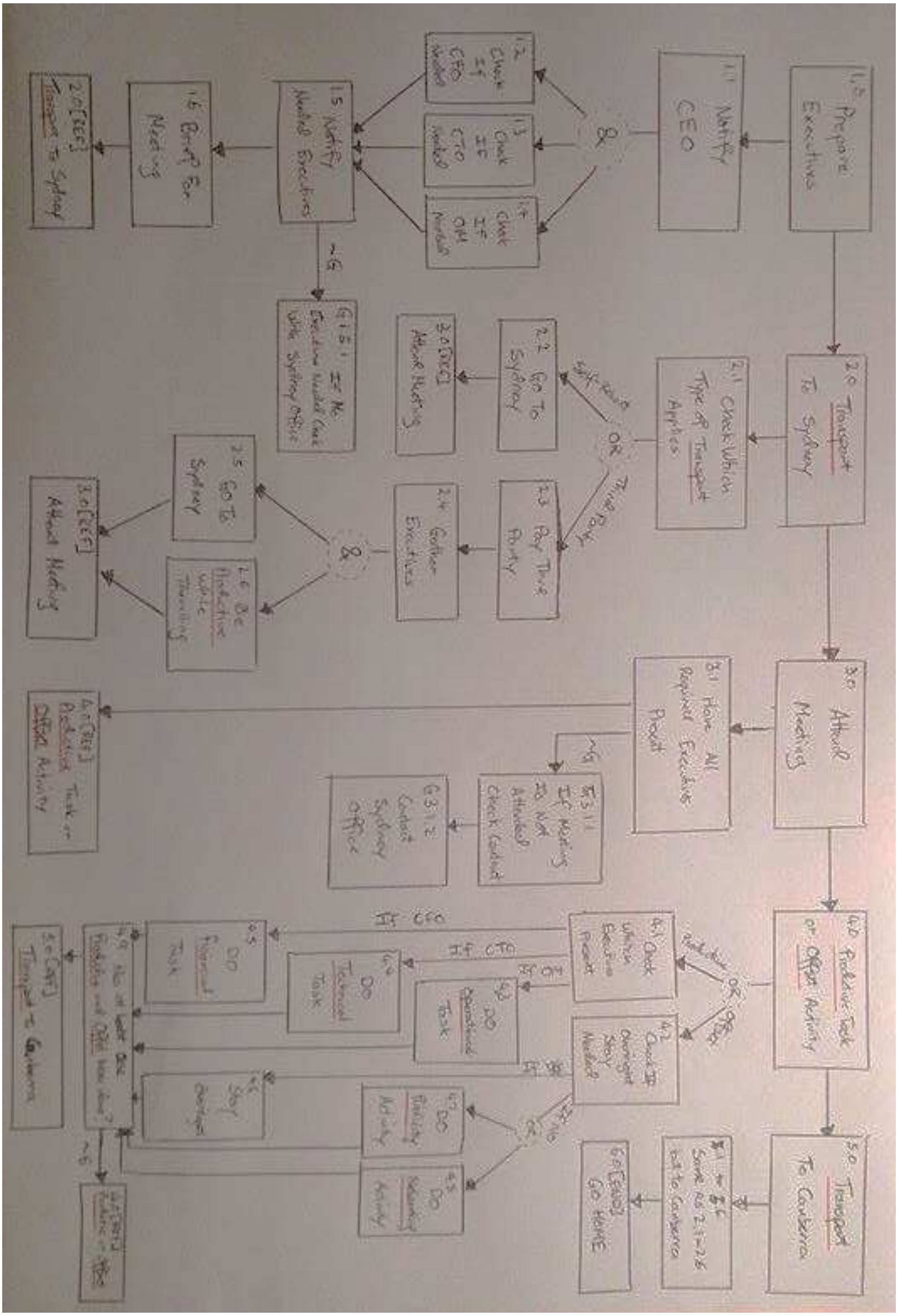
[4] L. Lin, “Subsystem Integration Individual Research Paper,” Canberra, 2014.

[5] J. Chen, “System Attributes Individual Research Paper,” Canberra, 2014.

[6] A. Zegelin, “Verification & Evaluation Individual Research Paper,” Canberra, 2014.

[7] J. Yu, “Design Communication Individual Research Paper,” Canberra, 2014.

Appendix 1 [1] [3]



Appendix 2 [1] [6]

Table A2a: Weighted Evaluation Matrix for Productivity (I)

Design Requirement	Weighting		Investments		Financing		Accounting		Product Inspections	
	Rank	Weight	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value
Practical	1	6	0	**0**	0	**0**	0	**0**	0	**0**
Appropriate	2	5	0	**0**	0	**0**	0	**0**	1	**5**
Time Efficient	3	4	3	**12**	3	**12**	3	**12**	3	**12**
Reliable	4	3	0	**0**	0	**0**	0	**0**	0	**0**
Convenient	5	2	0	**0**	0	**0**	0	**0**	3	**6**
Affordable	6	1	1	**1**	5	**5**	3	**3**	3	**3**
Totals				**13**		**17**		**15**		**26**
Scale	**5** – *Exceeds Compliance*		**3** – *Full compliance*	**1** – *Partial compliance*	**0** – *Non compliance*					

Table A2b: Weighted Evaluation Matrix for Productivity (II)

Design Requirement	Weighting		New Installations		Research & Development		Replenish Inventory		Visit Consultants	
	Rank	Weight	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value
Practical	1	6	0	**0**	0	**0**	0	**0**	0	**0**
Appropriate	2	5	3	**15**	5	**25**	1	**5**	0	**0**
Time Efficient	3	4	5	**20**	3	**12**	5	**20**	1	**4**
Reliable	4	3	0	**0**	0	**0**	0	**0**	0	**0**
Convenient	5	2	3	**6**	1	**2**	3	**6**	5	**10**
Affordable	6	1	5	**5**	0	**0**	1	**1**	1	**1**
Totals				**46**		**39**		**32**		**15**
Scale	**5** – *Exceeds Compliance*		**3** – *Full compliance*	**1** – *Partial compliance*	**0** – *Non compliance*					

Table A2c: Weighted Evaluation Matrix for Offset

Design Requirement	Weighting		Attend Conferences/Exhibitions		Marketing		Overnight Stays		Community Outreach	
	Rank	Weight	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value	Relative Compliance	Weighted Value
Practical	1	6	0	**0**	0	**0**	0	**0**	0	**0**
Appropriate	2	5	5	**25**	1	**5**	3	**15**	5	**25**
Time Efficient	3	4	5	**20**	1	**4**	0	**0**	5	**20**
Reliable	4	3	0	**0**	0	**0**	0	**0**	0	**0**
Convenient	5	2	5	**10**	0	**0**	1	**2**	5	**10**
Affordable	6	1	5	**5**	0	**0**	0	**0**	5	**5**
Totals				**60**		**9**		**17**		**60**
Scale	**5** – *Exceeds Compliance*		**3** – *Full compliance*	**1** – *Partial compliance*	**0** – *Non compliance*					