Final Report

Alexander Håkansson
19950617-1298, IT, 11h spent
alehak@student.chalmers.se

Kevin Hoogendijk
19950627-5594, IT, 8h spent
kevinhoog@gmail.com

FIRE Group 24
Mathematical modelling and problem solving

June 5, 2016

1. By submitting this report we confirm that the entire report is our own work, that it has been written exclusively for this course, and that we both actively participated in solving all the exercises of the course.

2. We will not spread our work in this course to others who take or can be expected to take the course.

1 Introduction

To solve great problems it will in most cases not be enough to just have a great set of tools and skills. Solving great problems also requires a strategy for problem solving and a broad and open view on the different ways problems can be solved. This report intends to summarize the insights into problem solving and modelling gained from the course in a way that can be helpful for others. This includes strategies we have learned to be successful, the essentials of different types of models, how these models can connect to eachother and more. These insights and experiences are a result of the course *Mathematical modelling and problem solving* at Chalmers.

Keep in mind that we both have a strong background in IT and as software developers, something that surely colors the way we think of modelling and their best usages. There could also be examples of application or implementation strongly tied to IT.

2 Different model types

Throughout the course it became more and more clear that it is much easier to even begin solving a problem if you can somehow classify it. However, the classification itself may not be trivial, especially if one does not have a broad idea of some different types of models. Learning about some different types of models, some standard problems and the solutions to these will be a great asset when tackling a new problem and has been a very important aspect of the course.

Before going into more detail about these it should be noted that a problem has no single unique way to be modelled, it is rather the opposite and most often you need to combine ideas from different modelling types. However, to be able to combine ideas you first need an understanding of the alternatives with their strengths and weaknesses.

2.1 Functions and equations

Functions and equations are probably what most people think of when they think of problem solving and mathematics in a combined sense. These types of models and problems are what most people with some science related education have probably spent hours upon hours constructing and solving, likely without actively thinking about the fact that they are doing mathematical modelling. Indeed functions and equations are somewhat of a foundation for all of the modelling types that are about to be mentioned, but it is important to note that modelling is not strictly bound to just functions and equations.

And what are these functions and equations? There are some very famous equations which do very well in modelling reality. Some of these are $E = mc^2$, $a^2 + b^2 = c^2$ and $F = ma$ - all of which do very well in modelling the real world. Equations do really well in problem solving since they are so concrete and all you need to do is to input some data and from there derive your answer, the problem is that there usually isn't a precise equation exactly for the problem you want to solve. This is where it starts getting limiting to think of modelling as only using functions and equations. Some problems would be extremely hard to solve if all you could do to try and solve it would be to sit down and try to find an equation for it.

Of course, some problems work particularily well with functions and equaitions. A good example would be physics problems since the same physical rules apply everywhere. However, when the problem is more specific to a constrained real world situtaion (like most problems that people will encounter) other modelling types will have to be considered.

2.2 Optimization models

Optimization brings a lot of interesting problems from real life scenarios and it lies in the nature of us to take the shortest path somewhere or spend as little or as much time as possible doing something. Anytime we have more than a couple of choices it gets hard to find the best solution manually. Many of the standard problems like the Traveling salesman problem are really old and has been used for ages. Getting into the age of computers, we now have much more power in our hands and it is easy to use algorithms to for example find the optimal paths between thousands of cities in a short period of time.

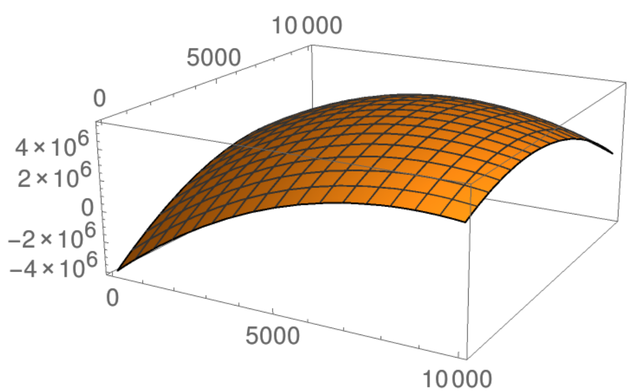

Figure 1: A plot of the profit over sold TV's

Optimization models are used a lot, it has helped us to develop society really fast. We can easily come to many great insights that will prevent throwbacks and enable profit. Consider a TV manufacturing company which has two models, 19" and 21". The profit of the 21" decreases for each 19" sold and vice versa. Without thinking of optimization, the manufacturer would probably try to create as many TV's that can be sold. This would possibly be a great mistake since the profit could turn negative. Modelling this as a linear optimization problem (seen in Figure 1) shows that manufacturing 10 000 of each model would result in lower profits than manufacturing and selling about half of each. Further study of this model also shows that if you were to manufacture more devices, you would get a negative profit.

Preventing these sorts of throwbacks are very important for us and for the development of the world. Even though the profit decrease probably can't be this precisely predicted and all of the variables are not easily identified, creating a model and optimizing it would give great insight in how to handle different situations. Which is something that goes for basically the whole course.

2.3 Dynamic models

A very powerful and widely used model type is dynamic models. In dynamic models change is a very large component, thereby the *dynamic* part. Usually it would be used to model change over time and considering that, it is not hard to understand why it is so widely used, especially in science and engineering. It is also not limited to model the change over time and usually involves the relative change between the involved factors.

Figure 2: A swarm of krill in the ocean

A rather easy to understand but powerful example that was brought up during the course is the so called *Whales and krill* problem which revolves around the relation between growth of whales and krill based on the population of eachother. It is basically a set of differential equations. The facinating and great thing about this is that if you just know how to model the problem you can easily simulate it and come to very profound and good conclusions about the problem - and this is a very good thing to do with dynamic problems; model them as one or more functions over time, simulate the model and look at the results and the relations between them.

An insight about dynamic models is that they are best used when simulated and then analyzing the simulations and the results to try to come to a conclusion about the problem. It is rather hard to use dynamic modelling to come up with pure algebraic solutions. This is likely since dynamic models quickly become very complex if you try to comprehend everything as a whole. It is better to break the problem down to smaller parts, define the relations between them and let a computer do the complex calculations for you. Considering this it is likely that the power and use of dynamic models will just grow as computers get more powerful - it would be very hard and expensive to simulate some more complex problems only 50 years ago which would be considered easy today.

2.4 Probability models

One may think that some things, like the weather for example, happen completely random with no relation to anything. Often the truth is that in that specific moment it might have felt random, but if you observe this during a considerable time interval you can gain enoguht data to model it. It probably has some connections to other things and these other things maybe gives us an explanation as to when this supposedly random thing will happen.

The probablilty of a weather phenomenon to occur, like rain, can often be found using statistics. We could also dig deeper into why it rains. The sun heats the sea which evaporates some water near the surface. This gas stack together and cools down creating clouds. Now rain is somehow dependent on how warm and sunny it is, which also has a probablity model. Putting many of these models together we can create great systems that could roughly predict the weather years in advance. We could also predict things with a very high certainity, which is very useful for many applications.

Pretty much everything in life has a probability of occuring, and this can be modelled if you have enough data or theory. Finding the probability of something occuring is increasingly important as limited resources needs to be prioritized in a continuously expanding society.

2.5 Discrete models

As the name suggest, discrete models are used for discrete problems. Because of their discrete characteristics these sort of problems are often found in computer science. This is something that many probably doesn't think about (at least if you don't have a computer science background). Like the fact that the web is basically a large graph.

The interesting thing about modelling real world problems as some sort of discrete model is that they get relatively easy to simulate and solve using a computer. There are also a lot of standard algorithms for solving discrete problems which means that as long as you make a good model you have to spend little time actually solving the problem.

Discrete models are not suited for everything though. You have a quite obvious restriction that the problem must be possible to model in a discrete way. This can be hard with a lot of real world situations. Suitable problems to apply a discrete model is when it for example is possible to model the problems as a graph. If you manage to model it to a graph in a way that covers the problem you have almost solved the problem already, since all you need to do is to apply some algorithm over graphs (and concidering that a lot of computer science revolves around graph theory there are plenty of great algorithms).

3 Application of modelling

Simply knowing about a lot of different modelling types will not be enough to actually apply them. It is also very important to realise that a problem can very rarely (if ever) be modelled as strictly one of the types mentioned in the previous section. It is required to combine, adjust and compromise between different types to get a good model of the problem if it is sufficeintly complex (which most real world problems are).

A problem that we encountered a lot mostly during the first half of the course was that we often tried to create a general purpose model for a given problem. This is probably a result from our background in software development where generalisation and abstraction is the one and only way to build complex software. What we learned was that when dealing with a real world problem that needs to be solved one should focus on solving the actual and precise problem that you have and not all possible combinations of the problem at the same time. This invovles using constraints, abstractions and assumptions that works for the given problem without caring about if they would work for a similar problem. This sort of focus is required to not get stuck in the modelling process.

3.1 Combination of modelling types

As mentioned, it is most often required to combine different modelling types to be able to model a problem or a real world phenomenon. This is not necessarily so easy if you don't understand the relations between the different types and which work well with eachother and which doesn't. We have gained quite some experience about this throughout the course.

A great example where modelling types mix is when you for example want to model some network of some sort and then optimize over it. It could be a computer network, activities spread out over time, locations or anything similar. These are discrete situations that could with some effort be modelled as a graph with some properties. When you have this graph you could perhaps apply some optimization strategies to find the best path through the network, or you could use some probability modelling to find the probabilities of different things to happen in the network. You could potentially even optimize over the probability to increase or decrease the probability of something happening.

When you are about to model something it is important to identify some properties so you know how to adjust the modelling types. If for example the problem involves some optimization it is necessary to adjust the model so that it enables for optimization solving. This is regardless of what other types of adjustment for different properties you have made, you can almost always adjust the model to fit a problem.

3.2 Modelling languages

In addition to only using and combining different types of models a very powerful way to do modelling is by using modelling languages. A modelling language is a sort of programming language, except it is often made for a set of specific types of problems. Modelling languages enables a more structured way to model and it can often be used to reduce the complexity of a model by smart language design.

An example of a modelling language that was mentioned in the course was the AMPL modelling language. The AMPL modelling language is a language for modelling linear optimization problems in a very well strucutured and powerful way. AMPL makes it easy to define complex constraints and to use large well structured data in the model. When the model get sufficiently large it would become unbearable to do all that modelling manually.

Another modelling language that we as software developers are very familiar with is the Unified Modelling Language (UML) as shown in Figure 3 which allows developers and engineers to define software in a unified and general way. This captures a very important aspect of modelling languages; the fact that they are general. What this allows is that anyone with an understanding of the modelling language could understand your model - this is very powerful in cases where you for example are about to turn an assignment over to someone else or if you need help in some way. Without a modelling language in this situation a lot of time would have to be spent by the other individual(s) just to understand how you are representing your model.

4 Problem solving

Problem solving has been a great aspect of this course and we have learnt a lot in this area. To create a good model is often not as straight forward as it seems, you might have to simplify and organize the problem before modeling it. Even when you have successfully

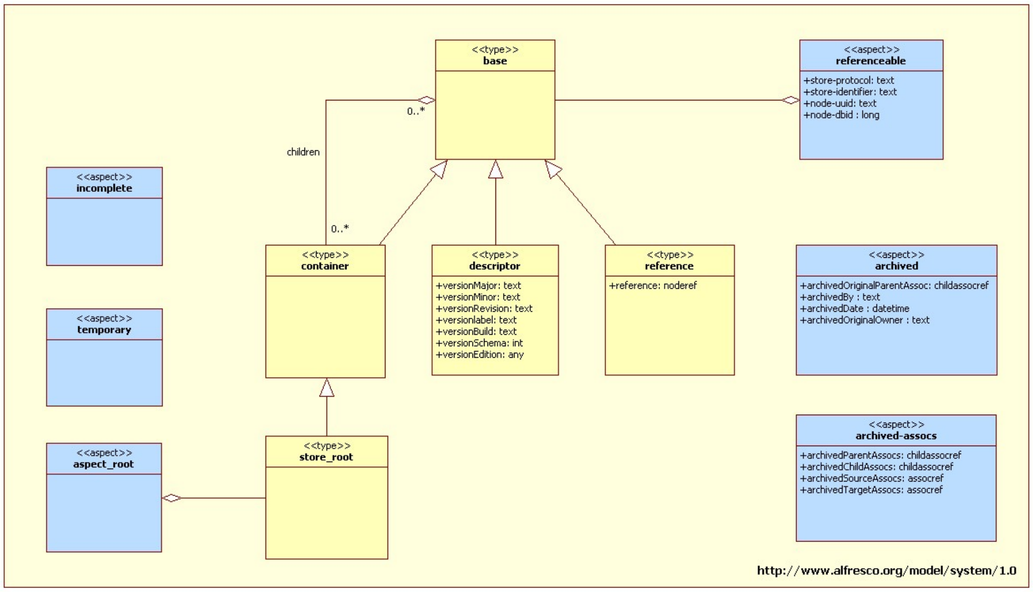

Figure 3: A simple UML diagram description

modelled a problem a problem solving strategy is still very important. You have to interpret what you get from modelling and come to conclusions on how to proceed. It is also always possible to improve the model in some way.

A common mistake that we realised is when you are too happy with a model too early, not considering that it may exist a better model for the problem. Putting some extra thought into a model often results in a better model. But this also goes the other way around. It is also important to not get stuck in the modelling procces. It is important to realise when the model is sufficient enough to go on and solve the problem. Throughout the course we have gained a lot of insights, gained extensive experience in how to do problem solving and what are some good tools and things to know.

4.1 Our improvement

At the beginning of this course during the first weekly assignment we were intrigued about how math was so much more than just doing calculations and equations. We didn't need it, we were just discussing and in the end came up with a model that fit the problem more or less. What was a huge block for us was that we kept getting stuck on unnecessary details. Instead of trying to consider all different details of the model we should be happy with what we got as an abstraction, learn from it, iterate and then create a model that is simply good enough to solve the problem.

Throughout the course we have been successfully identifying some of the problems we, as a group and as individuals, had. We have been encouraged to tackle these problems and that is exactly what we have done. The fact that even though the model is *good enough* there is probaly a better one, has made us more tenacious which in turn has made our models better. We got kind of an aha-feeling when we thought we had a good understanding of the *perfect* model for a problem and then someone else shows us a better one.

One interesting thing that we did in the end of the course, during the 5th weekly assignment, was to keep iterating over a model even though it was really good already and we knew this. We did this because we thought that there had to be a model that was more realistic. This made us get stuck for a while and in the end we actually ended up with a worse model than we already had. This is because we weren't happy with our model being a simplified and adjusted representation of the real scenario, but functionally it worked perfectly. What we learnt and improved from this is the fact that you should abstract away the realism as long as the model does what it is supposed to. And of course, if you are trying to create a model to simulate something realistic then this is what the model is supposed to do.

Altogether this course has broaden our way of thinking about problems and we are not as stubborn as we were in the beginning of the course. It feels like our problem solving has generally improved both considering speed and quality of the problems we solve and models we make.

4.2 General strategy

We have used our continously gained experience and insights to develop what we think is a really good general strategy for problem solving. One thing we found very helpful when starting off is to be well prepared. This might sound like something obvious but it's easily forgotten and not prioritized, but it is very important when it comes to problem solving or else a lot of time could be wasted. The way we do this is that we start the whole problem solving process with briefly discussing the problem ahead of when we are actually planning on sitting down and tackling the problem. If we do this with a day or two in between we might get some insights and a better understanding of the problem before we even start actively solving the problem. We see it as a way to subconciously process the problem.

When we then sit down and start with the problem solving process we have probably already avoided some common pitfalls by simply letting the problem sink in first. This part of the strategy was actually discovered totally by chance, one day we were discussing the next problem right before we were ending the session, and the next day we felt like we were really well prepared even though all we did was discuss the problem description.

We then discuss the problem in more detail and depth - maybe we also sketch up or try to simplify it to really get the whole picture. There are a lot of strategies/heuristics like this that can be applied to better understand a problem but we belive that the first step should almost always be to try to simplify the problem.

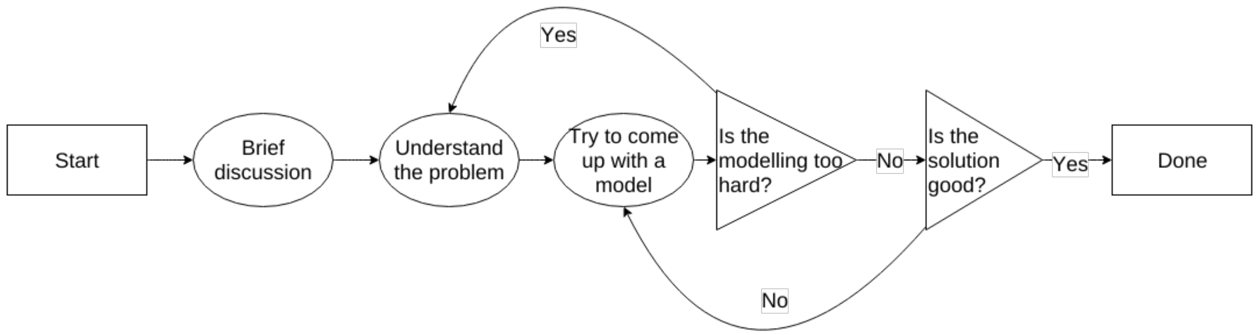

Figure 4: Our problem solving strategy

When we believe we have a good enough understanding of the problem we begin with the modelling. Since we now have a good understanding of the problem it is much easier to plan ahead and create a model that fits the situtaion very well. As mentioned before however it is important to not get stuck too long with the modelling. It is better to come up with a model that you think works and then try it out to see what happens. It is then rather easy to see what is faulty with the model and iterate over this process until you have a model that works. When the model is good enoguh it is just a matter of applying a good tool to generate a solution from the model. Most often when we are done it's a bit messy though, so an important aspect of the problem solving is to refine our solution and try to minimize it and make it as understandable as possible. If you do this with the approach that it should be as simple as possible for someone else to understand what you have done you will likely learn a lot yourself.

4.3 Useful tools

Although possible it is not very practical or cost effective to do all problem solving manually by pen and paper and there are a lot of different tools that can help in the process. As mentioned in the previous section there are lots of great modelling languages that can help to modularise and simplify modelling. As also mentioned there are a lot of different modelling types that can be combined and adjusted in powerful ways to create great models. But how do you actually come up with the solution given a good model? There are great tools for this, like Mathematica or MATLAB. These are programs that can help enormously in the pure computational and time consuming parts of the problem solving process.

In a way you need to let go of the imprinted thinking from school that you need to show all solutions from start to finish because when dealing with *real* problems this is either not practical nor necessary. If you can use Mathematica or MATLAB to save you maybe hours of manual work you really should. They could be used to brute force over possible solutions to find the best one, they can simulate and visualize things, they can determine relations between expressions and much more. When you realize that these great tools exist there is also an added aspect to the actual modelling: You need to create your model in a way that is friendly for a computer to work with - this is a whole process in itself which we believe requires some knowledge in how computers really work to be able to create optimized (fast) models.

Lastly, pen and paper is for most people considered old-faschioned but is actually a very good toolset in some situations. These days everyone think that the computer can solve everything, and it probably can. However, the problem has to be modelled so that the computer understand it. While trying to understand a problem it is extremely powerful to use this toolset to sketch and freely clutter around on a paper and this is extremely difficult using a computer. It is hard to see the computer replacing this in a near future.

5 Summary

One of the hardest parts of problem solving and modelling is to really understand the problem, and one can argue that you never really understand some problems fully. What is important is to be able to draw a line and make the decision to eventually try something out, so you don't get stuck too long. Even if you don't come up with the best solution straight away it is important to get started after analyzing for a while - this way you can try your thoughts out and improve on it.

To understand the problem is probably also the most important part of the problem solving process as a whole. If you suspect that you don't really understand the problem it is important to investigate more. The result of modelling and coming up with a solution without fully understanding the initial problem could be disasterous. The solution would likely not be correct and a lot of time would be wasted. This is something that we really take with us from the course and that we realise that we have started to implement in a lot of things.

Many things (if not everything) can be modeled if you understand them and can restrict or adjust the problem enough. However, doing this almost always results in a model that is not an exact representation of what you are trying to model and the result is rarely 100% realistic. The thing is though that from this model and result you can gain great knowledge about the problem or a result that is good enough for the application. And if the model or result is not good enough you just try to improve it and iterate - every faulty model you make gives more knowledge about the problem. When iterating over the model it will be more refined and better. It is great to know this during modelling so that you keep your motivation up, one faulty step is one step closer to the right solution.

The solution or final model can sometimes be very complex, it is important to have a good understanding of the solution as well as the problem. If not, there is a good chance that the solution you have doesn't truly solve the problem or the problem is more complex than what the solution can handle. We reason that if you can't describe your solution in a good and concise way so that someone else can follow it then you probably don't either have a good enough solution or you don't fully understand everything yourself yet.

The course itself has been very fun and has in many aspects improved our problem solving. We also feel that our way of thinking about problems has been widened a lot and it will probably help us much in the future. We would really recommend the course to anyone who whishes to be a better problem solver.