Utility Theory for Self-Adaptive Systems

T.J Glazier, Bradley Schmerl, Javier Cámara, David Garlan

December 2017
CMU-ISR-17-119

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Keywords: self-adaptation, utility theory

Abstract

Self-adaptive systems choose adaptations to perform on a running system. Typically, the self-adaptive ssytem must choose the "best" adaptation to perform in a given circumstance from a set of adaptations that may apply. Utility functions are typically used to encode some measure of goodness or badness the result of choosing a particular adaptation. Within the area of utility theory, there are a set of theories that could be used to help choose the best adaptation, that vary in the assumptions and requirements made about the system, the environment, and the business context of use. By understanding some of the formalities and advanced concepts in Utility Theory, engineers and administrators of self-adaptive systems can create more effective utility functions for their purposes, enable new forms of analysis, and potentially move into new frontiers in self-adaptive systems. In this report, we survey some of the more interesting topics in Utility Theory relevant to self-adaptive systems include objective and subjective expected utility, stochastic and fuzzy utility, and changing and state dependent utility functions.

Contents

Chapter 1

Introduction

Ever increasingly modern enterprise software systems are integrating self-adaptive capabilities into their production deployments to ensure the systems meet their defined quality objectives. Self-adaptation is the ability of a system to monitor its key metrics in the context of the environment in which it operates, and to change its configuration to perform better against its objectives. For example, a large scale web system might monitor its average response time and when it becomes too high it will add servers to handle the increased user load. In order for a self-adaptive system to work correctly it must have some way of determining if any given situation is "good", "bad", or some other measure of value. This required definition is encoded in what is commonly referred to as a utility function.

Utility functions, even within the self-adaptive community [10, 4, 6, 5, 3], have a long and rich history and deep body of knowledge. In fact, it can be reasonably argued that every adaptive system has a utility function within it. For example, a simple condition-action rule based adaptive system still require an architect or administrator to set the order in which the conditions and actions are applied, encoding a system of preferences. This system of preferences, as will be explained, can be captured in a utility function. By understanding some of the formalities and advanced concepts in Utility Theory, engineers and administrators of self-adaptive systems can create more effective utility functions for their purposes, enable new forms of analysis, and potentially move into new frontiers in self-adaptive systems. Some of the more interesting topics in Utility Theory relevant to self-adaptive systems include objective and subjective expected utility, stochastic and fuzzy utility, and changing and state dependent utility functions.

Subjective and objective expected utility has become one of the key tools in economics, game theory, and decision theory as its primary goal is to characterize decision making under risk and uncertainty. It is built upon a fundamental set of assumptions about how rational humans make decisions. This formal basis makes the theory attractive to self-adaptive systems engineers and administrators as it produces consistent and predictable results and its simplifying assumptions make the resulting applications tractable. Stochastic and Fuzzy utility are, currently, sparingly used in the context of self-adaptive systems because they are less predictable, potentially producing two or more different recommendations given the same scenario, but they more closely resemble human decision making. However, they have great potential in several different contexts including security in which a measure of unpredictability in the behavior of the adaptive system could potentially be an asset in defending itself. For example, an attacker doing surveillance of a system will be much more challenged if the system is presenting dynamic and uncertain behavior. Changing and state dependent utility are used to consider additional factors in the construction of utility functions including how the utility might be effected over time and how the utility function might change depending on specific factors.

The areas of subjective and expected utility, stochastic and fuzzy utility, and changing and state dependent utility theory are only a portion of the vast body of knowledge that constitutes utility theory. This report focuses on these areas as they are the ones most relevant to individual self-adaptive systems. Several other areas of utility theory might be applicable to collections of self-adaptive systems and are briefly covered in the overview of additional topics of utility theory (section 2).

This technical report will describe in relevant detail subjective and expected utility (section 5), stochastic and fuzzy utility (section 6), changing and state dependent utility functions (section 7) in the context of an exemplar scenario (section 3).

Chapter 2

Overview of Additional Topics in Utility Theory

Utility theory is a vast body of knowledge with a long history of serving as the foundation for a number of different areas of study. [1] and [2] provide a comprehensive survey and distillation of this corpus. However, this report focuses on the areas of utility theory that are most directly applicable to a single self-adaptive system, but this section will give a brief overview of the other areas of utility theory and comment on the potential applicability to the area of self-adaptive systems.

1. **Expected Utility in Non-Cooperative Games** - This is an area of utility theory that develops the mathematical framework of preferences for choosing amongst different strategies in a non-cooperative, often adversarial, game. Overwhelmingly, it makes the assumption of trying to understand how players *should* choose their strategies and not how they *will* choose them. This is done by making the assumption that the players are either experts in the field themselves or are being advised by one. This assumption is popular in the application of game theory for social sciences and economics. In self-adaptive systems, this area of study can have some application in understanding the relationship between the environment and the adaptation manager or between two adaptation managers which are attempting to, in some way, fight against the other. This has substantial potential application in the area of self-securing systems. [2, p. 979]

2. **Expected Utility in Cooperative Games** - A counterpart to the theory in non-cooperative games, this area addresses the utility theory that underpins coalitional behavior, including games with and without side payments and "transferable utility". For the area of self-adaptive systems, this area is relevant to a coalition of self-adaptive systems and how they can potentially work together to maximize the utility for each individual member. As modern IT systems are more increasingly being made up of multiple adaptive systems, this area is a significant potential area of active interest. [2, p. 1065]

3. **Utility in Social Choice** - This portion of utility theory describes how decisions should be made to maximize utility to a significant number of people or individuals, like a society. For example, this area would focus on the utility theory of elections and individual societal issues. Similar to objective and expected utility, this areas makes a number of assumptions regarding "rational" behavior that are considered idealistic in human societies and based upon those develop an interesting result: a collective decision is not possible. The most likely way in which a decision can be made in a society is by "dictatorial" means. While this is an oversimplified description of the theory, it might have substantial applicability for collections of autonomous systems where the assumption of 'rational' behavior is more likely to hold. [2, p. 1099]

4. **Interpersonally Comparable Utility** - Unfortunately, utility does not have a standard scale or units of measure. Each individual will develop their own scale and measures along with their utility functions. Therefore, it is necessary to understand when individual measures of utility can be compared

against each other and what valid analyses and conclusions can be drawn. This is most relevant to collections of self-adaptive systems. For example, if you could compare the utility measures amongst a group of self-adaptive systems it could be determined which systems is ‘most in need’ and how much ‘more in need’ that system is compared to the others which can facilitate a potential collective response. [2, p. 1179]

Chapter 3

Exemplar System

One of the barriers preventing a practical understanding of utility theory and its application to self-adaptive systems is the complexity and opaqueness of the mathematical theory. To facilitate the practical application of utility theory, the following exemplar system will be used to demonstrate how the various mathematical constructs might be utilized in an autonomous system [9].

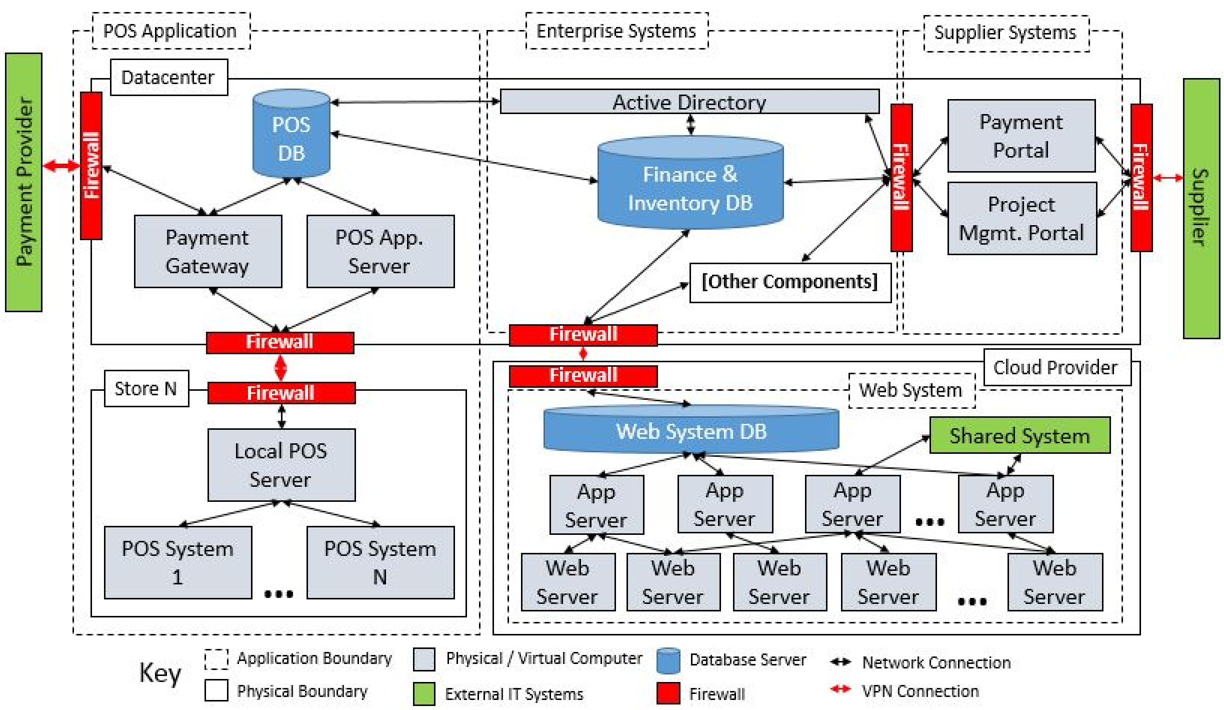

Figure 3.1: Exemplar System Architecture

This architecture is representative of a large brick-and-mortar retail chain with many geographically distributed stores, a large web presence, and integrations with various B2B partners, suppliers, and contractors.

The web system is a large scale cloud deployment that utilizes at least one autonomous manager to elastically scale the size of the infrastructure given the current demand on the web site. For example, the web system will have a much larger deployment during cyber-Monday than on Christmas morning. The use of multiple autonomous managers might be required to individually scale the various tiers of the architecture or individual subsystems of the web application (e.g., the shopping cart vs. the product catalog). Overwhelmingly, the system architects and administrators are concerned with the average web page response time or the time it takes for an incoming user request to receive back the generated web content. Consequently, this is the primary concern of the autonomic manager. The autonomic manager has three tactics available to it:

1. **Add Server** - This tactic will add a new computing unit, typically a virtual machine, to the front end web server array as a way of increasing capacity to handle additional user load.
2. **Eliminate Complex Content** - This tactic will cause the web system to not serve back to clients any heavy weight graphics or other enhanced content to try and reduce the size of the web page payload and the response time.
3. **Traffic Shaping** - This tactic will cause the network for the web system to prioritize high value traffic and potentially ignore low value traffic to try and reduce the web user load.

In general, the architects and administrators prefer the autonomic manager to add a server ahead of reducing complex content which is ahead of traffic shaping. This is because while adding a server does increase costs, it preserves the integrity of the experience for all of the users. However, the system cannot add servers indefinitely, therefore the other options are available but less preferred. Each of these tactics has the potential to directly impact the primary metric under management, the average web page response time.

The in-store point of sale (POS) systems are fixed physical hardware deployments, but the number of transactions processed in a given period of time can vary greatly given various environmental conditions (e.g., Black Friday). However, it is essential to the health of the business that average transaction time per customer checkout remain as low as possible. Therefore, it is critical for the payment processing infrastructure (e.g., the payment gateway and POS application server) to be dynamic to handle the changing transaction volumes. Additionally, this infrastructure is subject to both denial of service and data exfiltration attacks due to the value of the data and the importance of the system to the enterprise. Therefore, the autonomous system manager will need to constantly monitor and adjust the infrastructure to detect and eliminate potential threats. Similarly to the web system, the POS adaptation manager is concerned with not only average transaction response time, but also with maintaining a high transaction reliability rate, often 99.99% or better. The following are the tactics available:

1. **Add Server** - This tactic will add a new computing unit, typically a virtual machine, to the processing array to try and handle increased transaction load.
2. **Switch Payment Gateways** - This tactic will switch the provider responsible for authorizing credit card charges in the event that their system is unreliable or non-responsive.
3. **Black Listing** - This tactic will add a particular user's ip address to a black list if it is unknown or producing abnormal transaction volumes to mitigate potential rogue terminals issuing a DDoS.

For the POS system the architects and administrators will also generally prefer to add a server over switching payment gateways over black listing users. However, these tactics are often deployed in reaction to different stimuli in the environment. For example, if the POS system is experiencing high transaction volume then add server is likely to be the most appropriate response, problems with the payment provider will result in a switch of payment gateways, and finally black listing a connection is typically in response to a detected DDoS attack or other security incident.

Finally, the supplier systems will not experience high variability in demand, but they are a tempting target for data exfiltration attacks because of the type of data they contain, but also because of the integrations with "soft target" suppliers and vendors. This is also similar to internal employees with remote system access over a VPN presenting another type of 'soft target'. The autonomous manager of these systems will be concerned with detecting and analyzing interactions to discover anomalous behavior and eliminate the risk. The autonomic manager for the supplier systems might have the following tactics available:

1. **Kill Connection -** This tactic will eliminate a connection which is experiencing unusual behavior like connections at odd times of day or other unusual usage patterns.

2. **Revalidate User -** This tactic will challenge a user to prove they are a human and potentially have them re-authenticate to the system.

Unlike the other autonomic managers for the web system and the POS system, the autonomic manager for the supplier system is not concerned with key metrics like performance or reliability, but instead it attempts to actively monitor and mitigate security risks. Therefore, the architects and administrators preference will be for ‘no action’ meaning that they only want the system to act under unusual circumstances and then the preference is to revalidate the user ahead of killing the connection outright.

Chapter 4

Definitions of Preference and Utility

Overly simplified, the purpose of a self-adaptive manager is to make a choice amongst various adaptation alternatives. Given an unacceptably high web page response time, should the system add servers? Turn off complex content? Some combination of the two? Or possibly do nothing at all. Each of these options has pros and cons and given any specific situation the system must determine which of the options is most preferred and least preferred. Mathematically this ordering is known as a preference relation. However, as will be seen, the definition of preference relations are quite complex and generally not convenient for analysis. This is where the utility function comes in. The utility function provides a convenient mathematical representation that solves this problem by providing a score, a measure of goodness, that simplifies the analysis of the options, but it must be understood when the utility function will accurately represent the preference relation and when it will not. It is important to understand that the preference relation and the utility function representing it are actually two separate and distinct things.

While the first explicit introduction of the distinction between preference and utility appears in the *Manual of Political Economy* by Vilfredo Pareto in 1906 and even though he recognized the difference between the two, he did not realize that determining the ordering of preferences in a principled manner was actually a non-trivial problem. For example, what does it mean for one alternative to be preferred over another? Can you be 'indifferent' between two alternatives? and many more questions.

Adaptive systems must establish preference relations as part of their functionality to select the most appropriate adaptation option. Often it is the intent of the adaptive system architect and administrator to have the adaptive system mimic a human decision making process and select the option the human would select if presented the same choices. Therefore, it is critical to understand that while the framework and definitions on the preference relations might seem innocuous, they actually start to impose a definition of 'rational' behavior which may not mimic a human decision making process. For example, if a human prefers apples over bananas over oranges then a human should prefer apples over oranges regardless of the option of a banana. While this preference relation and consequence of violation of it might seem trivial, applied to adaptive systems the situation and consequences can be more abrupt. For example, an architect might prefer adding a server over eliminating complex content over traffic shaping. However, if there are no more servers to add then the administrator might prefer traffic shaping over eliminating complex content. This is actually defined as 'non-rational' behavior because given the same environmental conditions the human had two different preference orderings. While generally understandable for humans, this can be quite problematic for the formal guarantees of adaptive systems. Establishing all of the required considerations which influence these decisions, especially in humans, is a practically impossible task. Therefore, a series of assumptions need to be made to establish the framework under which formal guarantees can be made and narrow the possibilities to consider to make application of the theory practical.

To begin to solve this problem, Wold in 1944 was the first to construct a set of rules which precisely defined what a preference relation is. The mathematical bedrock on which preference relations, and ultimately utility theory, are built is through a set of common set theory properties, which we define below, but in the interest of brevity has been simplified. For a full treatment please see [1, p. 2]

Definition 1. *A relation R on a set X is said to be:*

- ***Reflexive*** *if xRx for all x in X*
- ***Symmetric*** *if xRy implies that yRx for all x, y in X*
- ***Transitive*** *if xRy and yRz imply that xRz for all x, y, z in X*
- ***Connected*** *if for all x,y in X either xRy or yRx*

R is any relation between two objects, not necessarily a preference relation, that will be defined in a moment. The reflexive property asserts that every item (e.g., adaptation option) in the set (e.g., all possible adaptation options) has a relation with itself. Symmetric asserts that if one option has a relationship to another, then the converse is also true. Transitive asserts that if one option, y, has a relation with two different objects, x and z, then x and z also have a relationship between them.

Definition 2. *A relation R on a set X is an* ***equivalence relation****($\sim$) if it is reflexive, symmetric, and transitive.*

Therefore, in the context of utility theory, if you wish to establish a preference relation amongst a set of alternatives, all three of those properties, reflexive, symmetric, and transitive must be true. In the context of adaptive systems this means that two options, adding servers and eliminating complex content, are equally preferred to each other. If you can define when two options are equivalent, you can also describe when one is preferred over another.

Definition 3. *A* ***preorder*** *($\precsim$) on a set X is a reflexive transitive relation on X.*

Definition 4. *A connected preorder on a set X is called a* ***total preorder****.*

These definitions give a natural representation to the relationships of preferences. For example, given the two alternatives of adding servers, x, and eliminating complex content, y, then if $x, y \in X$ then $x \sim y$ if and only if $x \precsim y$ and $y \precsim x$. Meaning that if adding servers is better than eliminating content, and vice versa, then the decision maker does not care which one is actually chosen. Precisely stated, the relation $\sim$ is then an equivalence relation on X meaning that a decision maker will be indifferent between the two alternatives. Additionally, if $x, y \in X$ then $x \prec y$ if and only if $x \precsim y$ and $\neg(y \precsim x)$. Meaning that in this case adding servers is absolutely preferred over eliminating complex content. The relation $\prec$ is called the strict part of $\precsim$ and a decision make will always prefer x to y given those two options.

Definition 5. *A* ***utility function****, f, is an order preserving function, an order-homomorphism, on a set of alternatives, X, endowed with relation R.*

A utility function is an equation that preserves only the order of the preferences amongst each other. For example, if adding servers, x, is preferred to eliminating content, y, then the utility function might assign a score of 0.9 to x and 0.3 to y which preserves that order assuming highest score wins. However, the scores of 0.11 for x and 0.10 for y would give the exact same result and meaning. Nothing can be said about the difference between the scores. For example, in the first scenario, x is not three times more preferred to y, it is simply preferred. After some other mathematical assertions not directly relevant to adaptive systems, there is a formal guarantee that if the preference relation follows those rules then a utility function is guaranteed to exist. This constitutes a significant step, but it does not provide any any further guarantees about the utility function itself. For example, is it continuous and differentiable? These properties are important to adaptive systems as they provide the guarantees for the validity of multiple types of analysis that you can perform with the utility function. For example, if an administrator wanted to evaluate the change or rate of change in utility over time as a measure of the effectiveness of the adaptive mechanisms the utility function would need to be continuous and differentiable to enable this analysis.

Mathematically these properties can be asserted if the adaptive system architect can make one of two different assumptions. The first, is that there is always an option that is available to the adaptive system. This assumption, known as strong or weak monotonicity, ensures that there is at least one option (e.g., adding servers) that the administrator will prefer because it is as good as any other (weak monotonicity) or one that is better than the others (monotonicity). The second possible assumption, known as non-satiation, is that there will always be an option that is more preferred. For example, an adaptation manager will always prefer a faster response time or an individual will always prefer more money. These two assumptions are related to each other. Specifically, non-satiation implies monotonicity, but not the other way around. These assumptions and formalisms have specific implications for adaptive systems.

Definition 6. *Let $\precsim$ be a total preorder on a subset X of $\mathbb{R}^N$. Then the preorder is said to be* ***weakly monotonic*** *if $x < y$ in $\mathbb{R}^N$ implies that y is not strictly worse than x or equivalently if $x \precsim y$. The preorder is said to be* ***monotonic*** *if $x < y$ in $\mathbb{R}^N$ implies that x is strictly worse than y, i.e., $x \prec y$.*

In the context of adaptive systems, this ensures that there is at least one option (e.g., adding servers) that the administrator will select because it is as good as any other (weak monotonicity) or one that is better than the others (monotonicity). It is important to note that in the context of adaptive systems 'no action' is a potential alternative. Because that option will always be available, either because it is as good as any other or strictly preferred, implies that we will be able to construct a continuous and differentiable utility function. However, if the context permits, the non-satiation assumption is mathematically and practically stronger and produces utility functions of more analytical interest.

Definition 7. *A preference relation $\precsim$ on a subset X of $\mathbb{R}^N$ is said to be* ***locally non-satiated*** *if for each $x \in X$ and each open ball $B(x, r)$ with center x there is $y \in B(x, r)$ such that $x \prec y$.*

This non-satiation assumption asserts that in an arbitrarily small region, there is always another preference option that is more preferred. In the context of adaptive systems this is seen in an administrators preference for key metrics (e.g., web page response time) to always be better. In economics, this is also apparent in that people always prefer more money and this serves as the mathematical basis for the economic concept known as *marginal utility*, an area of potential interest in adaptive systems. For example, with a non-satiation assumption there will be a point of diminishing return in which trying to further satisfy the preference will no longer yield the same return.

To better understand these mathematical concepts and their impact on adaptive system administrators, we will expand upon the exemplar scenario and construct a few potential utility functions. Assume there are only two tactics available: add servers (x) and eliminate complex content (y), and there are two metrics that the system monitors: response time (r) and cost(c), and that the system is evaluating options at the point in time in which r is greater than some configured value for max response time. Then the following are some possible utility functions:

1. $x \succ y$ if $c < c_{max}$
2. $y \succ x$ if $r_{new} < r_{old}$

Both of these are trivial examples to be sure, but they are representative of the various mathematical properties that have been established so far. For example, in option 1 the function preserves the preference ordering of adding a server over eliminating complex content as long as the maximum cost is not exceeded. Because this criteria can be satisfied this function does not meet the criteria for non-satiation and as such is not differentiable, but is a valid utility function. However, option 2 presents a set of criteria in which there is always a more preferred option, lower response time, as such it can be differentiated and a point of diminishing returns potentially established.

In the landscape of adaptive systems, these two examples are representative of the most common types of utility functions in use for adaptive managers to make choices given multiple options. One of the goals of representing choice in adaptive systems is to try to match the decisions that human administrators would make in the same conditions. However, the mathematical framework requires specific assumptions about how humans make decisions that might not actually be true. For example, a mountain climber certainly prefers the alternative of 'life' to 'death' and given any probability of 'death' rationally should not participate. However, people still climb mountains. The problem here is that there are other factors that influence the decision, like the thrill of making it to the top. An example in adaptive systems might involve two systems, X, Y, who are exchanging resources based upon some current state of the environment(Env). If X is sensitive to the fact that Y is an enterprise critical system, system X might prefer that no resources be transferred regardless of the state of the environment and X's preference might decrease with the amount to be transferred. Therefore,

$$\begin{gathered} Env_1 \text{ causes } X \text{ to transfer 5 resources to } Y \\ \text{and} \\ Env_2 \text{ causes } Y \text{ to transfer 10 resources to } X \end{gathered}$$

X may exhibit the following preferences:

$$(2/3Env_1, 1/3Env_2) \succ (1Env_1, 0Env_2) \succ (0Env_1, 1Env_2)$$

This pattern would violate both monotonicity and non-satiation as X prefers Env_2 when it occurs by chance as opposed to having it outright. While the example is a bit artificial it does demonstrate that if the underlying problem domain has some of the more common human decision patterns, like pleasure and enjoyment or self-sacrifice, then some of the mathematical assumptions fail to hold endangering the principled construction and analysis of utility functions.

As utility theory and its associated uses (e.g., in economics) are principally concerned with the study and representative modelling of human behavior, this is a significant source of criticism for the theory. However, adaptive systems have one advantage. The assumptions about how decisions are made are more likely to hold as they can be controlled in the construction of the system and there is less direct human decision making. The inevitable problem will come when the architect of an adaptive system must construct the utility function to either 'mimic the human action', in which case the mathematical assumptions are unlikely to hold, or 'perform the best action', which might not align to what a human would do. This conflict often arises in the context of adaptive systems, especially in self-protecting systems where a human might choose to 'leave a minor threat in the system to study it' while the autonomous system's best action is to 'eliminate the threat'. Therefore, adaptive system architects and administrators need additional formalisms to enable this type of reasoning (e.g., stochastic and fuzzy utility) in addition to the more common approaches to adaptive system utility functions (e.g., expected utility).

Chapter 5

Objective and Subjective Expected Utility

Handling risk and uncertainty is the primary motivation for the creation of adaptive systems. Therefore, the expected utility hypothesis, which characterizes decision making under uncertainty and risk, has become a key theory in adaptive systems. However, the primary problem with characterizing reward under risk and uncertainty is that there is no guarantee of any particular result or reward at any given point in time. For example, if the adaptive system decides to add a server, but there is a technical problem preventing it from coming online, the risk and uncertainty has prevented the reward. Therefore, mathematicians have used the 'expected value' of a random variable determined over a large number of trials. For example, if the adaptive system chooses to add a server 1000 times and, on average, it results in an improvement in response time of 10%, known as the expected value, then that is the value used even though the actual number might vary in any given trial. This system of trials is often referred to as a lottery.

However, it was widely observed that humans often do not consider potential alternatives over a large number of trials. This contradiction was frequently known as the difference between the 'mathematical expectation' versus the 'moral expectation'. In the context of an adaptive system an architect might have the choice to configure an adaptation manager to take a 1/80 gamble on the chance of improving overall system performance by 10% or nothing. The expected value of this gamble is 10% / 80 = 0.125%, the 'mathematical expectation'. However, system architects would generally choose a guaranteed payoff of a 0.1% improvement, the 'moral expectation' of the gamble, despite the fact that it is clearly suboptimal.

This general contradiction was widely acknowledged and accepted by mathematicians until the mid-twentieth century when von Neumann-Morgenstern provided an axiomatic basis [7] and established the expected utility hypothesis, which provides a mathematical framework that addresses this contradiction. Interestingly, addressing this contradiction was not their primary motivation. von Neumann and Morgenstern's work on expected utility was a 'minor diversion' from their primary work in game theory. However, since then any utility function whose value is to be maximized has been called a von Neumann-Morgenstern utility function (NMUF). The following is an abbreviated version of the complete mathematical framework, a full treatment can be found in [1, p.143].

Definition 8. *Let Y denote an arbitrary set of possible consequences. A **simple lottery**, λ, is a probability distribution mapping $\lambda : Y \to [0,1]$ with the property that $\sum_{y \in Y} \lambda(y) = 1$. Let $\Delta(Y)$ denote the set of all such simple lotteries.*

In the context of adaptive systems this establishes the mapping for each of the potential consequences of an action, creating a lottery. For example, if the choice is to add a server then the potential outcomes might be a 1%, 5%, 10%, or 15% improvement in the response time for the web system. Each of these potential outcomes has a probability associated with it. For example, 0% improvement is a 1% probability, 5% is a 10% probability, 10% is an 80% probability, and finally 15% is a 9% probability. Combining these potential results weighted with their probabilities gives a compound lottery defined here:

Definition 9. *Given any pair $\lambda, \mu \in \Delta(Y)$ and any $\alpha \in [0,1]$ define the convex combination, or* ***compound lottery****, by: $[\alpha\lambda + (1-\alpha)\mu](y) := \alpha\lambda(y) + (1-\alpha)\mu(y)$ for all $y \in Y$.*

Applying this definition to the previous example, the percentage of improvement can be expected is: (0.01 x 0.01) + (0.05 x 0.10) + (0.10 x 0.80) + (0.15 x 0.09) = 0.0986 or 9.86%. Given the uncertainty in the environment and the potential outcomes, the expected value of this particular lottery is a 9.86% improvement in the response time for the web system. Determining the expected value of the lottery is required to determine what the best choice or adaptive option might be. This makes compound lotteries a fundamental and common part of the analysis in adaptive system managers. Examining the compound lottery for each potential action does not actually determine which one is 'best' or preferred. All it does is provide the expected outcome of choosing that decision. To determine the 'best' one, you need a point of comparison that can be used to evaluate all of the options. In mathematical terms, this is known as the degenerative lottery.

Definition 10. *Given any consequence $y \in Y$, let $1_y \in Y$ denote the* ***degenerate simple lottery*** *in which y occurs with probability 1. Then each simple lottery $\lambda \in \Delta(Y)$ can be expressed in the form:*

$$\lambda = \sum_{y \in Y} \lambda(y) 1_y$$

For self-adaptive systems the degenerate lottery is the guaranteed result the adaptive system could achieve. For adaptive systems, in general, it is implicitly assumed to be the 'do nothing' action, the result of making the choice to not take any adaptive action. The degenerative lottery sets a baseline to determine if the potential lotteries are preferred amongst each other. Fortunately, the specific definition of the degenerative lottery value in any given context is not critical to the overall mathematical framework as long as the same value is used in all comparisons. Therefore, most commonly, the value of the current known state is used for the purposes of comparison in adaptive systems. The framework for expected utility to this point has provided a score by which comparisons amongst choices can be made, but has not yet defined the actual method of comparison. For example, does the high score win or the low score? While seemingly a trivial question it is important as it sets another expectation regarding the behavior of the decision maker, that they will universally want to make things as good as possible. In humans, a potentially problematic assumption but in self-adaptive systems a potentially reasonable one.

Definition 11. *A* ***feasible set****, F, is any set in a domain, D, of non-empty subsets of a given underlying set or choice space Z. For each $F \in D$, the choice set C(F) is a subset of F, and the mapping $F \mapsto$ C(F) on the domain D is the choice function.*

This defines a convenient mathematical construct to define the expected utility hypothesis. Specifically, that there is a non-empty set of potential lotteries (e.g., adaptation choices) to choose from (F) and the choice function ($C(F)$) (e.g., which choice is preferred) is the method of determining which choice is preferred.

Definition 12. *Given a von Neumann-Morgenstern utility function $v : Y \to \mathbb{R}$ such that, given any feasible set $F \subset \Delta(Y)$, the* ***choice set*** *is:*

$$C(F) = \underset{\lambda}{\operatorname{argmax}} \left\{ \sum_{y \in Y} \lambda(y) v(y) | \lambda \in F \right\}$$

This gives the formal definition for the choice function, but more directly, $C(F)$ consists of those lotteries $\lambda \in F$ which maximize the expected utility function defined by:

$$U(\lambda) := \sum_{y \in Y} \lambda(y) v(y)$$

This defines the expected utility hypothesis, which is overwhelmingly the preferred method of constructing utility functions within adaptive systems. Simply stated, it says that if you have the scores for lottery of each of the choices, then the option with the highest score is preferred. Note that, in a formal sense, this still does not give any indication about how much a choice is preferred over another. It does not matter if the difference between scores is 0.5 or 0.1, one is simply preferred over the other. However, as a matter of practicality in adaptive systems, scores that are reasonably close together (e.g., with a range of 0.05) are often considered equivalent to each other. As convenient, and in practice simple, as this mathematical theory is there are several implied properties that this mathematical framework mandates that are quite controversial and might not hold or be desirable in every context for self-adaptive systems.

The first of these is the ordering condition. This mandates a transitive relation on the scores of the compound lotteries themselves.

Definition 13. *The Ordering Condition(O) is defined as:*

$$\lambda \succsim \mu \iff \sum_{y \in Y} \lambda(y)v(y) \geq \sum_{y \in Y} \mu(y)v(y)$$

The primary problem and critique on this condition is that intransitivities often occur when evaluating potential choices. For example, when humans are presented with pairs of decisions and asked to state their preference they will often make a set of decisions that leads to inconsistencies. For example, a human might prefer apples to bananas, bananas to oranges, and oranges to apples. In a self-adaptive system this might look like the system preferring adding a server to eliminating complex content, eliminating complex content to traffic shaping, and traffic shaping to adding a server. It is actually reasonably straight-forward to find situations in adaptive systems in which pairwise comparisons can lead to these inconsistencies. In order for your expected utility analysis to be mathematically sound, these intransitivities cannot occur. In the context of adaptive systems, if these did occur, you could not guarantee that the adaptive system will make the same decision given the same set of circumstances and depending on the context that could be problematic.

Another source of intransitivities, equally relevant to adaptive systems, is that the comparisons between objects are often incomparable alternatives which provoke ‘responses’ on several different ‘attribute’ scales. Although each scale might be transitive, the combination of them might not be. For example, given the following alternatives in the web application of the exemplar system:

Option 1: Service Quality: +10%, Cost: +12%
Option 2: Service Quality: +5%, Cost: +4%

Assuming you want to maximize service quality and minimize cost, option 1 is the preferred option for service quality, and option 2 is preferred for cost. Which one should be chosen? While each scale, quality and cost, is transitive the combination of them is not which results in potentially inconsistent behaviors. For architects of autonomous systems, it might be possible to construct mathematical functions to combine the dimensions to result in a transitive behavior. In the above example, it is possible to define a quality to cost ratio in which option 2 is preferred but this is highly context dependent and care must be taken to guarantee the transitivity within the constructed function. But regardless of the actual source of intransitivities it is generally accepted they will exist in many decision spaces and this is a known tradeoff of the theory in order to produce mathematically tractable results.

The second of these conditions is the independence condition which asserts that the preference for any two options is independent of the presence of a third option. For example, a human might prefer apples to bananas, but if oranges are available then a human might prefer oranges over bananas over apples. In this situation the preference between apples and bananas changes because of the presence of oranges.

Definition 14. *The Independence Condition (I) is defined as: Given* $\lambda, \mu, \nu \in \Delta(Y)$ *and* $0 < \alpha < 1$*, then:*

$$\lambda \succsim \mu \Longrightarrow \alpha\lambda + (1-\alpha)\nu \succsim \alpha\mu + (1-\alpha)\nu$$

This statement of the independence condition is due to Samuelson in 1983, but is not the only definition. There are at least 4 different definitions each with varying conditions that make the condition stronger or weaker. This particular condition presents the most significant source of criticism for the notion of expected utility and the strongest challenge to its practical use. In the context of autonomous systems, this would say that the choice of any two adaptations is independent of the existence of any given third option. However, this often does not hold. Given the example of an autonomous system maintaining a quality of service on the exemplar system web site the following three adaptations might be possible:

Option A: Add Server
Option B: Remove Complex Content
Option C: Turn On Traffic Shaping

If the autonomous system has only option A and B available, it might order them in that way in order to maximize the users experience with the web site. However, if option C is available, the system might order them as C (most preferred), B, A because if they have already eliminated the low value traffic, then adding servers, and increasing cost, may no longer be an effective option. It is easy to construct examples in both human and autonomous system decision problems that would violate this independence condition. This is another area where it is generally accepted that inconsistencies will exist in favor of mathematical tractability, but that is also why there are multiple versions of this condition with differing strengths. Each of them has different consequences for the mathematical analysis that can be conducted, but no practical difference between them in the context of adaptive systems. In the interest of brevity the nuances and variations are not examined here, please see [1, p.213] for a full treatment. As with the ordering condition if the constructed utility function does violate this condition then the predictability of the actions of the autonomous system cannot be guaranteed.

In the context of adaptive systems this results in two potential ramifications. First, in the situation in which one tactic is not available (e.g., no more servers available to add) then the utility function must order the other preferences the same way under the same conditions. Otherwise, the adaptation manager will actually be unpredictable. Second, if a new tactic is developed (e.g., D: to prefer paying users over non-paying users) then the adaptive architect must revisit the preference relationship and, consequently the utility function, as the combinations of tactics and their availability / unavailability might result in a violation of this independence condition.

The third condition of note is the continuity condition. It states that given two choices there exists a point in which the individual making the decision is indifferent between the two choices.

Definition 15. *The Continuity Condition(C) is defined as: Given $\lambda, \mu, \nu \in \Delta(Y)$ with $\lambda \succ \mu$ and $\mu \succ \nu$ this requires that there exist $\alpha', \alpha'' \in (0,1)$ satisfying:*

$$\alpha'\lambda + (1-\alpha')\nu \succ \mu \text{ and } \mu \succ \alpha''\lambda + (1-\alpha'')\nu$$

Overwhelmingly, the continuity condition is widely accepted under the provision that the decision choices are not extremes in the decision space. For example, it would violate the continuity condition to chose a payoff of $0.01 vs. $1.00 and death in a particular lottery as there is no point at which you are indifferent between the two options. Fortunately, these types of extreme decision spaces are quite rare or the probability of the extreme consequence, death, is so low as to constitute a practical impossibility. In the context of adaptive systems this particular condition presents little practical trouble for the architect or administrator of an adaptive system, but might be relevant in extreme security scenarios where the adaptation might be to shut down the system or crash the UAV. It is included here as all three of them are necessary to guarantee that a von Neumann-Morgenstern utility function can be constructed with maximum value of 1 and a minimum value of 0.

Up to this point the expected utility theory has made an assumption that has not been made explicit. Specifically, the probabilities of each individual consequence are objectively known. Returning to a previous example where 0% improvement is a 1% probability, 5% is a 10% probability, 10% is an 80% probability, and finally 15% is a 9% probability, it is assumed that these probabilities are either known or their values are assumed to be known. This is known as objective expected utility. But what if they are not known? This is known as subjective expected utility and is a supplemental body of knowledge in utility theory.

In the context of the exemplar system, specifically the POS system, it is possible that there are different 'states of the world'. For example we might define three states based upon how many sales are being made at a given time as low, medium, and high. Each of these potential states of the world can lead to a different set of probabilities for each of the potential consequences of the choice. For example, if the POS system is experiencing a high state of sales then the choice to add a server might have a 10% chance of having no effect on the system, when it is in a medium state of sales it might have a 5% chance, and in low possibly a 1% chance. In the context of adaptive systems subjective expectedly utility is an enhancement to expectedly utility which allows you to take into account changes in the environmental context to evaluate and determine the correct preference ordering.

Formally, let Y be a fixed domain of possible consequences and let S be a fixed set of possible states of the world. $y^s : S \to Y$ is a contingent consequence function(CCF) and $y^S = \langle y^s \rangle_{s \in S}$ is a list of such functions.

Definition 16. *The Subjective Expected Utility (SEU) function is defined as:*

$$U^S(y^S) := \sum_{s \in S} p_s v(y_s)$$

where p_s are non-negative subjective utilities or different states $s \in S$ satisfying $\sum_{s \in S} p_s = 1$, $v : Y \to \mathbb{R}$ is a NMUF.

The subjective expected utility hypothesis, similar to its objective counterpart, determines that the decision maker will choose to maximize the subjective expected utility function. Also similar to the objective expected utility function, there exist a set of conditions that must also be met for subjective expected utility to apply.

The same independence, ordering, and continuity conditions apply but they now also apply to the state of the world. For example, an administrator would prefer a low or medium amount of sales traffic in the POS system as opposed to a high sales volume. The independence condition needs to be enhanced with a new restriction known as the Sure Thing Principle (STP). STP formally says that the preference relation for the states of the world is independent of any event that is not included in the list of potential states. For example, in the POS system the administrators preference for low or medium sales traffic vs. high is independent of any other external factors like the potential consequence of full system shutdown.

A new state independence(SI) condition must also be established. For example, it does not matter to the POS system if the high sales volume is due to a slashdot effect (e.g., Black Friday) or a DDoS attack against the system, the result is a high state of sales volume. STP and SI might seem similar and to some degree they are, but they define independence on two different things; one is the consequences of the state of the system (high, medium, low sales volume) and the other is on the state of the world itself (slashdot effect or Black Friday sales).

Finally, there is the ordering of events condition (OE) which states that the individual probabilities of the individual states of the world occurring induce a preference ordering for those states of the world. For example, if the probability of high sales volume being the result of black friday is 75% and a DDoS sttack is 25%, this implies that the administrators would prefer the black friday state of the world to a DDoS.

The breadth of options, required conditions, criticisms and counter examples can certainly lead an adaptive systems architect or administrator to believe that the proper construction of a NMUF to be practically impossible. It is actually quite commonly achieved in adaptive systems and in other fields like economics and game theory. Overwhelmingly, practitioners tend to simplify and abstract the problem and impose assumptions sufficient to meet the requisite demands. However, these abstractions and assumptions should be done conscientiously with the background knowledge. Unfortunately, this is often not done which can lead to adaptive systems demonstrating potentially unpredictable behavior. As adaptive systems move into more life- and safety- critical applications like self-driving cars and IoT solutions, this could have significant potential consequences. Finding this abstraction and appropriately combining the factors to meet the expectations of the theory is a significant and important challenge for the adaptive systems architect.

Chapter 6

Stochastic and Fuzzy Utility

As detailed in previous sections, traditional utility theories make a set of simplifying assumptions to ensure the tractability of the mathematical framework. Broadly, Stochastic and Fuzzy Utility refer to theories of preference that do not specifically try to impose specific assumptions about how the decisions should be rationally made. Instead they take a more pragmatic approach and attempt to create a mathematical framework that does not seek to explain human decision making, but instead seeks to emulate it. Returning to a previous example, according to expected utility theory climbing a mountain, even with its small chance of death, should never be selected as a potential action, but people still do. Stochastic and Fuzzy utility do not eliminate the possibility of someone selecting that option, they simply assign it a lower probability that an individual would select it given a set of weekend activities available. This approach would better mimic human decision making as the framework would not predict that some people would go mountain climbing, but given the same conditions on two consecutive weekends the choice may not be the same, leading to a degree of unpredictability and potentially not choosing the 'optimal' behavior.

When modeling decision behavior as a probabilistic process it becomes clear there are multiple parts of the process that could exhibit some type of random-like behavior. Since, overwhelmingly, these processes are unobservable (e.g., they are occurring in a person's head, in the environment, or in a third party system) there is no clear way to develop a common mathematical framework. Therefore, Stochastic Utility is actually a collection of methods of assigning probabilities to different and multiple parts of the mathematical framework. This allows an adaptive system to act in a manner that can be inconsistent or appear to be random, mimicking the way a human might make decisions. Each of the methods below are a different way of assigning probabilities to each of the preferred choices instead of outright selecting the most preferred option. Therefore, given the same set of conditions and circumstances, an adaptive system using stochastic and fuzzy utility will not always select the most preferred option, it will choose an adaption strategy with some degree of randomness. This could potentially be useful in some adaptive system contexts (e.g., security) where some degree of randomness and unpredictability is beneficial (e.g., to make it harder for an attacker to drive adaptations in attack-friendly directions).

Fortunately the individual representations use a common base vocabulary and model to ease definitions and analysis. Let $\mathcal{A}$ be the set of potential objects of choice (e.g., the set of adaptation options), P be the probabilities of preference relations (e.g., the set of probabilities for the adaptation options), p denote probabilities of individual choices (e.g., the individual probability of selecting each option), $\mathcal{R}$ be the set of disjoint preference relations on $\mathcal{A}$ (e.g., all of the possible combinations of choices), and $P(R)$ for $R \in \mathcal{R}$ be the probability of a particular preference relation R (e.g., the probability of preferring a specific strategy) where $\sum_{R\in\mathcal{R}} P(R) = 1$. Choice probabilities, p, are then defined on a set $\mathcal{C}$ of ordered pairs (A, B) of subsets of $\mathcal{A}$ with $A \subseteq B$ and $B \neq \emptyset$ and $p(A, B)$ is the probability that an individual will chose an alternative or object in A when the choice is restricted to set B. For notional convenience, in a binary decision the full notion $p(\{x\}, \{x, y\})$ for $x \neq y$ will be abbreviated to $p(x, y)$.

The first set of representations are examples of binary choice probabilities where for all $x, y, z, w \in \mathcal{A}$:

R1: $p(x,y) \geq 1/2 \Leftrightarrow x \succsim y$, where $\succsim$ is a weak order on $\mathcal{A}$

This is possibly the simplest representation available in stochastic utility. It states that if the probability of choosing x is greater than $1/2$ then x is preferred to y. For practical application, it is certainly possible to redefine the threshold for various circumstances. For example, an adaptive systems architect might set the threshold for taking adaptive action at $2/3$ to ensure that action is only taken when it is deemed highly necessary. In the context of the supplier portion of the exemplar system the adaptive architect might set a $2/3$ value for maintaining the connection when suspicious activity is detected, but a $1/3$ chance of killing the suspicious connection. This would have the intended effect of some suspicious connections being terminated while others are allowed to stay.

R2: $p(x,y) \geq p(z,w) \Leftrightarrow u(x) - u(y) \geq u(z) - u(w)$ where $u : \mathcal{A} \rightarrow \mathbb{R}$

This representation works on the strength of the difference between the two options. The larger the difference in the scores, u, between the two options the higher the probability that option will be selected. An adaptive systems architect might prefer this method as it works to promote decisions that are not 'close calls'.

R3: $p(x,y) = \varphi(u(x) - u(y))$ where $u : \mathcal{A} \rightarrow \mathbb{R}$ and φ is a nondecreasing function from $\mathbb{R}$ into $[0,1]$ for which $\varphi(0) = 1/2$

This representation directly assigns a probability for the selection of an option based upon the difference between the scores, u, of the two given options with an equal score yielding a $1/2$ probability. This representation is attractive to an adaptive systems architect who wants to set the probability of a given choice being made to be directly proportional to the strength of the difference between the two options. For example, if the users activity in the supplier system is highly suspicious the adaptive system administrator can set the utility function such that the preference for killing the connection yields a much higher score leading to a higher probability it will be chosen.

R4: Given $u : \mathcal{A} \rightarrow \mathbb{R}$ define u on B as:

$$u(B) = \sum_{x \in B} u(x)$$

$p(x,B) = u(x)/u(B)$, where $u : \mathcal{A} \rightarrow \mathbb{R}$ and u is strictly positive.

This representation sets the probability of selecting any individual choice from a set of choices as being directly proportional to the total score of the other available options. An adaptive systems architect would prefer this option as it directly relates the probability of an option to its score proportionally to the total score of the other options, but it must be understood that this simply sets a probability and does not guarantee that the highest scoring option will be selected.

Representations R1 and R2 are often referred to as ordinal representations because they involve only binary choice probabilities and are partial representations. Representations R2-R4 have been called constant utility models for choice probabilities because each uses a single u with a fixed value $u(x)$ for each object.

In contrast to the preceding models which assign probabilities to choices depending on preferences, a random utility model uses a set U of $u : \mathcal{A} \rightarrow \mathbb{R}$ along with a probability distribution μ for selecting a member of U. This model generally assumes that not all members of U will produce the same ordering as $\mathcal{A}$. In practice this means that the adaptive system architect defines a set of utility functions, each of which sets a preference ordering over the set of choices. Then you assign a probability distribution to the set of utility functions which will pick one of them with a degree of randomness. Once the utility function is selected the system will then work to maximize the choices against that function. In the interest of brevity the presentation of random utility models is intentionally simplified to ignore some of the more formal mathematical notions that have no direct impact on applicability, a full treatment can be found at [1, p. 273].

Formally, a random utility representation of choice probabilities is provided by a probability space (V, μ) in which V is the set of all $\mu : \mathcal{A} \to \mathbb{R}$ and μ is a probability measure on V. V_{AB} is then defined for each $(A, B) \in \mathcal{C}$ of all $u \in V$ for which $x \in A$ maximizes utility in B:

$$V_{AB} = \{u \in V : u(x) \geq u(y) \text{ for some } x \in A \text{ and all } y \in B\}$$

Then the primary random utility representation for choice probabilities is, for all $(A, B) \in \mathcal{C}$:

R5: $p(A, B) = \mu(V_{AB})$, where (V, μ) is a probability space

This model is the most generic version of a random utility model, but more specifically and more commonly the following model is used:

R6: $p(x, B) = P(\mathcal{R}(x, B))$, where P is a probability distribution on the complete ranking domain $\mathcal{R}$ and $\mathcal{R}(x, B) = \{\succ \in \mathcal{R} : x \succ y \text{ for all } y \in B \backslash \{x\}\}$

Practically, these models both determine the probability of a given choice as being the probability of selecting a utility function that would prioritize that specific choice. Random utility models present a unique capability to adaptive systems architects to allow systems to periodically and randomly choose their current priority through selection of their utility function. This presents interesting implications for the broad topic of continuous improvement or proactive adaptation as well as specific security applications. For example, assume that an adaptive system architect does not want the system to merely respond to a set of conditions, but instead continuously evolve its adaptation patterns to try and find an optimal set. However, if the utility function is constant then the system might be stuck in a local minimum/maximum. By randomizing the selection of the utility function, which is directly responsible for the creation of the adaptation patterns, the system can search for a most optimal set and evolve. Further, adding unpredictability to the selection of the utility function and changing the adaptation patterns can dramatically hinder a malicious attackers attempts to drive behavior to exploit or create a security vulnerability.

As the presented models demonstrate, Stochastic utility provides a number of different methods of assigning probabilities to any individual choice. However, fuzzy utility takes a different approach. Instead of assigning probabilities to a particular choice for random selection, it seeks to establish a strength of preference between all combinations of choices and then select the "best" one depending on desired criteria. For example, if you have a set of potential adaptation options like add server, eliminate complex content, and traffic shaping then fuzzy utility will pair each of these options to each of the others and select the best one depending on defined criteria. Assume the combination of add server receives a score of 0.5 and eliminate complex content gets a score of 0.3. Additionally, the combination of add server gets a score of 0.6 and traffic shaping gets a score of 0.5. Should the 'best' choice be the combination with the highest absolute score or the highest average score between the two choices? Or should it be the lowest absolute score or the lowest average score, or a number of other potential options?

Formally, a fuzzy binary preference is a relation over a set of choices, X, is a function r from $X \times X$ to $[0, 1]$. This relation, r, can have a variety of properties depending upon how the relation is constructed. For example, r has reflexivity if for all $x \in X, r(x, x) = 1$ and irreflexivity if for all $x \in X, r(x, x) = 0$. Other potential properties include min-max transitive, mix-transitive, mean-transitive, pos-transitive, bin-transitive, and several others that are not specifically defined in the interest of brevity. However, the definition of the desired criteria for making the "best" choice has a more direct practical impact.

Each of the following models is a method of selecting a particular option and use a basic short hand notation to distinguish the method of selection.

Suppose $x \in B$:

$$
\begin{array}{ll}
MF(x,B) = \max\limits_{y \in B-\{x\}} r(x,y) & mF(x,B) = \min\limits_{y \in B-\{x\}} r(x,y) \\
MA(x,B) = \max\limits_{y \in B-\{x\}} r(y,x) & mA(x,B) = \min\limits_{y \in B-\{x\}} r(y,x) \\
SF(x,B) = \sum\limits_{y \in B-\{x\}} r(x,y) & SA(x,B) = \sum\limits_{y \in B-\{x\}} r(y,x) \\
MD(x,B) = \max\limits_{y \in B-\{x\}} (r(x,y) - r(y,x)) & mD = (x,B) = \min\limits_{y \in B-\{x\}} (r(x,y) - r(y,x)) \\
SD(x,B) = \sum\limits_{y \in B-\{x\}} (r(x,y) - r(y,x)) &
\end{array}
$$

where “M” is max, “m” is minimum, “F” is for, “A” is against, “S” is sum, and “D” is difference.

As a practical example, the MF or Max-For model will select the set of choices with the maximum preference score from $X \times X$, while the SA or Sum-Against will select the set of choices with the largest sum against them. In each case, the strength of preference between the two options is calculated by the r relation as previously defined. For an adaptive systems architect fuzzy utility presents distinct advantages. Specifically, what this procedure does is compare all combinations of choices against each other and given the preference for the method of comparison (e.g., Max-For or Sum-Against, etc.) chooses the best one. This direct comparison allows for the most nuanced and direct comparison of the potential alternatives as opposed to something like expected utility which attempts to make a selection by evaluating the potential outcomes of the choices and scoring them without regard for how one directly compares to another. This form of analysis can also result in transitive inconsistencies (A $>$ B $>$ C $>$ A) which expected utility would not. However, the size of the decision space $X \times X$ can present a computational challenge in near real time applications depending on the number of choices.

Stochastic utility and fuzzy utility both attempt to not make any types of assumptions about how decisions are actually made, but instead simply try to provide mathematical tools which mimic the often conflicting and non-rational process that humans frequently exhibit. Therefore, the adaptive systems architect must be aware that utilizing these techniques in a practical system can yield some quite dramatic behavior that would be highly unlikely from a “rational” method. For example, in a safety critical situation, like the automatic breaking system on a car, it is possible that the system chooses to speed up because of the assignment of a probability to that choice in stochastic utility or that the choice to speed up vs. slow down yields a higher preference score because the trip will not take as long in fuzzy utility. While this is a contrived example for sure, it demonstrates the inherent complexity for the adaptive systems architect in mimicking human behavior or navigating the complexities and inconsistencies in rational behavior.

Chapter 7

Changing Utility Functions and State Dependent Utility

All of the previous utility models made the simplifying assumption that utility was being evaluated instantaneously at any given point in time without regard for how the utility function might change over time. In general, there are three types of changes that can effect utility functions: (1) changes with respect to intertemporal substitution, (2) changes with respect to substitution between different commodities, and (3) changes in respect to substitution between alternative lotteries. In the context of self-adaptive systems these equate to modelling the utility of the current and future state of consuming a particular commodity, like the budget available to run the system, the utility of consuming and substituting multiple different commodities like fuel and electricity in a car, and finally the utility of consuming a particular commodity like budget with uncertainty in the state of the environment.

The first of these models is attributed to Samuelson and is commonly referred to as the discounted utility model. This type of utility formulation would be useful to a system administrator if they wanted to ensure that the system considers the potential future states of the system and how it is allocating various resources. This could be useful for an area like the web application to understand if it would be better to spend the resources now to expand the capacity of the system or to wait. His formulation has the following representation:

$$U(c_1, c_2, ..., c_T) = \sum_{t=1}^{T} u(c_t)\delta^{t-1}$$

where $C = (c_1, c_2, ..., c_T)$ is the consumption profile or the amount of the commodity consumed at each time interval, $u(c_t)$ is the instantaneous utility derived at time period t from consuming c_t, and δ^t is the discount factor, $0 < \delta < 1$. Further it is assumed that $u' > 0$ and $u'' < 0$. In this model, the discount factor plays a significant role in determining the importance of the future state utility in the present calculation. For the adaptive systems architect, the larger the value and the more the future utility state is taken into account the more likely it is to affect the current course of action with a frequently changing environment. However, the lower the value the more the system will simply consider the present state without regard for the potential future state of the utility. Commonly this model is used to compare alternative consumption profiles to determine which might yield the greatest utility and there is a continuous time counterpart represented by:

$$U(c) = \int_{0}^{T} e^{-pt} u(c(t))dt$$

Both of these models assume additive separability amongst the various arguments meaning that the evaluation at each time period is not influenced by the ones before it. This has served as a significant source of criticism for the models but is widely used as it dramatically simplifies the analysis.

While the preceding model allowed for the evaluation of utility over multiple periods of time, Pollack developed a model that allowed for the analysis of the consumption of different types of commodities. Returning to the exemplar system, for an adaptive systems architect this could be the use of monetary resources used to keep the system online and the amount of network bandwidth available. Let p_i be the price of commodity i, x_i the quantity consumed of the i commodity, and M the total expenditure (budget) available. The utility function takes the form:

$$U(x_1, x_2, ..., x_n) = \sum_{k=1}^{n} a_k ln(x_k - b_k)$$

where for all k, $a_k > 0$, $x_k > b_k$, and $\sum a_k = 1$. An interesting property of this model is that the utility becomes infinitely negative as x_k approaches b_k. For this reason b_k is commonly referred to as the subsistence level or the minimal level required of that commodity for that particular domain. For humans this might be something like water, for a car this might be the minimum fuel level for the current journey. With this the model is logarithmic for all levels above subsistence. This means that the consumer will first allocate $\sum_k b_k p_k$ of the budget, M, then allocates the residual $M - \sum_k b_k p_k$ which maximizes the logarithmic utility. This makes the demand function for the i-th commodity:

$$x_i = b_i - \frac{a_i}{p_i} \sum_k b_k p_k + \frac{a_i}{p_i} M$$

Further extensions to this model allow for the notion of 'habit forming' in which the subsistence level varies in a linear fashion based upon the previous consumption level, $b_{i,t} = b_i^* + \beta_i x_{i,t-1}$ where b_i^* can be thought of as the "physiologically necessary" component and $\beta_i x_{i,t-1}$ as the "psychologically necessary" component. For an adaptive systems architect this is an important result as it allows for analysis of the consumption of commodities over time. This allows one to determine if the commodity demands, x_i, t will converge to a steady state level or not and analyze the potential short term and long term behavior of utility. For example, if a car consumes the fuel now for acceleration that might result in a short term gain with a long term deterioration in utility.

Finally, the third extension handles uncertainty in the state of the environment. This allows the POS and web application of the exemplar system to examine potential futures to determine what the correct allocation of resources would be. Based upon the model for consumption over time it is represented as:

$$U(c) = \sum_{t,i} \delta^{t-1} p_i u(c_{t,i})$$

where $c_{t,i}$ is the consumption at date t conditional on the occurrence of event i, p_i is the probability attached to event i, and $\delta^{t-1} p_i$ is the discount factor on the utility, $u(c_{t,i})$. For the adaptive systems architect this model presents the ability to perform analysis on the utility given potential states of the environment and their probability of occurring. While generally beyond the scope of adaptive systems, these three models form part of the basis for the mathematical representation of prospect theory as they show much of the dynamic inconsistency in the human behavior observed by Kahneman and Tversky [8].

Chapter 8

Conclusion

Utility theory and self-adaptive systems have a long and rich history together. In fact, it could be argued that each self-adaptive system is fundamentally built upon utility theory. Therefore, it is important for architects and administrators of self-adaptive systems to have an understanding for both the theoretical and practical basis for utility theory in working with self-adaptive systems. Some of the more important things to understand are:

1. **Preference vs. Utility** - Architects and administrators encode a set of preferences into the self-adaptive system for it to make appropriate choices in its context. However, it is important to understand that the preferences and the utility function are actually two different things and great care must be taken in the construction of the utility function to ensure it accurately and consistently reflects those preferences.

2. **Utility Function Always Possible** - In the context of self-adaptive systems, as long as 'no action' is available, then it is always possible to construct a utility function for the self-adaptive system. However, this does not provide any guarantees about how the utility function will behave.

3. **'Rational' Behavior or Human Behavior** - As presented, there are two different approaches to how self-adaptive systems can make decisions. The first is the idea of 'rational' behavior which defines conditions to ensure consistency in the decision making process. Given the same circumstances, the same decision will be made. The second approach is the idea of mimicking human behavior in which significant inconsistencies and some measure of 'randomness' ensues. Given the same circumstances, it is not guaranteed that the same decision will be made. Both of these approaches have significant relevance in self-adaptive systems.

   The applicability of the various forms of utility theory, in particular objective and subjective expected utility and fuzzy and stochastic utility, is important for administrators and engineers of self-adaptive systems to understand as their proper use will aid in the creation systems that meet their objectives.

Bibliography

[1] Salvador Barber, Peter Hammond, and Christian Seidl. *Handbook of Utility Theory. Volume 1 Principles.* 01 1998.

[2] Salvador Barber, Peter Hammond, and Christian Seidl. *Handbook of Utility Theory. Volume 2 Extensions.* 01 2004.

[3] Nelly Bencomo, Amel BELAGGOUN, and Valrie Issarny. Dynamic decision networks for decision-making in self-adaptive systems: A case study. In *ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, 05 2013.

[4] Shang-Wen Cheng, David Garlan, and Bradley Schmerl. Architecture-based self-adaptation in the presence of multiple objectives. In *ICSE 2006 Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, Shanghai, China, 21-22 May 2006.

[5] Daniel Cunha da Sliva, Adilson Barboza Lopes, Filipe A. P. Pinto, and Jair C. Leite. Selecting architecture configurations in self- adaptive systems using qos criteria. 12 2012.

[6] Camilo Fitzgerald, Benjamin Klöpper, and Shinichi Honiden. *Utility-Based Self-Adaption with Environment Specific Quality Models*, pages 107–118. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[7] Oskar Morgenstern John von Neumann. *Theory of Games and Economic Behavior.* 05 1944.

[8] Daniel Kahneman and Amos Tversky. Prospect Theory: An Analysis of Decision under Risk. *Econometrica*, 47(2):263–291, March 1979.

[9] Ryan Wagner, Matthew Fredrikson, and David Garlan. An advanced persistent threat exemplar. Technical Report CMU-ISR-17-100, Institute of Software Research, Carnegie Mellon University, July 2017.

[10] W. E. Walsh, G. Tesauro, J. O. Kephart, and R. Das. Utility functions in autonomic systems. In *International Conference on Autonomic Computing, 2004. Proceedings.*, pages 70–77, May 2004.

List of Figures

List of Tables