9
Generating Explanations to Justify Choice

In Chapter 6, we proposed a decision making technique that incorporates principles adopted by humans to make decisions, in order to facilitate extracting a rationale behind decisions and produce explanations for choices made. We have also identified in our study of how explanations can justify choices the kinds of explanation users expect to receive in order to understand and accept a decision made by a software system on their behalf. We now propose in this chapter an explanation technique that bridges the gap between these two approaches — we generate explanations that meet identified requirements based on the reasoning process of our decision making technique.

Before detailing our explanation generation approach, we introduce in Section 9.1 the notation adopted in this chapter, and remind the reader of structures of our reasoning technique, used in our explanation approach. Then, we show in Section 9.2 how we select attributes to be part of explanations, which are parameters of the explanation templates presented in previous chapter. As different explanations can be given to justify a choice, we describe in Section 9.3 how to select the type of explanation to be given and also how to put the selected parameters together with the explanations, showing how to generate explanations. Section 9.4 shows an example of generated explanations for our apartment example. Finally, we compare our approach with existing work and present a performance evaluation in Section 9.5.

9.1
Notation

Our explanation generation approach follows the same notations adopted in Chapter 6, and is based on many of the structures produced in our decision making technique. o_c is an option chosen from the set of those available, Opt, and the remaining ones, i.e. $Opt_r = Opt - \{o_c\}$, are rejected options o_r. Options are characterised by a set of attributes, Att. Variables o and a, possibly with an index, represent an option and an attribute, respectively. We also have a set of modifiers M, which consists of expressive speech acts (e.g. *like*, *love*, *hate*, *don't accept*) and rates (e.g. *good* and *bad*), and these are used to express monadic preferences. Some of these modifiers indicate hard-constraints to be considered in the decision making

process. Finally, we recall structures built during the execution of our decision making technique, which are used to generate explanations.

- $PSM[o, a]$: Preference Satisfaction Model (PSM) is a partial mapping from an option and an attribute to a modifier (or its negation). It represents how the value of an option attribute is evaluated in terms of a modifier.
- $OAPM[o_i, o_j, a]$: Options-Attribute Preference Model (OAPM) is a mapping from a pair of options and an attribute to {+, −, ∼, ?}. It shows if the attribute value of an option o_i is better (+), worse (−), as good as (∼) the value of the same attribute of an option o_j. If there is no information available, the OAPM value is "?".
- $d(o_i, o_j) = (1 - w_{to} - w_{ea}) \times Cost(o_i, o_j) + w_{to} \times ToContrast(o_i, o_j) + w_{ea} \times ExtAversion(o_i, oj_2)$: the decision function represents how much o_i is negatively evaluated with respect to o_j. It can be seen as the cons of o_i (w.r.t. o_j), as opposed to $d(o_j, o_i)$, which are its pros. $d(o_i, o_j)$ is the weighted sum of three factors: the costs provided by attribute values ($Cost(o_i, o_j)$); and the trade-off contrast ($ToContrast(o_i, o_j)$) and extremeness aversion ($ExtAversion(o_i, oj_2)$), which are two principles that humans adopt when making decisions (Simonson and Tversky 1992).
- $Cost(o_i, o_j) = \sum_{a \in Att} w_a(o_i) \times AttCost(o_i, o_j, a)$: the costs of o_i with respect to o_j are captured by a real value [0, 1], which is calculated as the weighted sum of the cost provided by each individual attribute. Attribute weights may vary for each option, as there may be priorities expressing conditional importance of attributes.

9.2 Explanation Parameters: Selecting Relevant Attributes

Explanation patterns presented in Chapter 8 give templates for explanations, which are parameterised by a single attribute, or sets of attributes. In this section, we show how these attributes (or this attribute) are selected to be part of the explanations. We will follow the same order adopted in the previous chapter to present patterns.

9.2.1 Single-attribute Selection

The first explanation pattern consists of identifying an attribute that is critical for making the decision. An attribute is critical when it is the reason for choosing a particular option, and the values of other attributes are not relevant. In addition, as described in the Critical Attribute pattern, there may be constraints — used

to express extreme cases that users do not accept — but they are satisfied by all options. A critical attribute is identified based on the OAPM, as its values associated with the chosen option are +. For the remaining attributes, there are two possibilities for the OAPM value: (i) ∼, which indicates that constraints were given, but the other options also satisfy them; and (ii) ?, meaning that no preference was given with respect to a particular attribute. The definition of *critical attribute* is presented below, which indicates that if there is an attribute that is critical, it is unique.

Definition 9.1 *Let o_c be the option chosen from the set Opt by a decision making technique. An attribute $a_{crit} \in Att$ is said the* ***critical attribute*** *of the decision, or $CriticalAttribute(o_c)$, if for all options $o_r \in Opt_r$, we have $OAPM[o_c, o_r, a_{crit}] = +$, and for all the other attributes $a \in Att$ and $a_{crit} \neq a$, o_c is considered as good as o_r ($OAPM[o_c, o_r, a] \neq \sim$) or there is no preference given over this attribute, i.e. $OAPM[o_c, o_r, a] \neq$ "?".*

In many situations, people have hard-constraints, which eliminate options, whose at least one attribute value is non-compensatory, that is, it is not possible to compensate this value, regardless the values of other attributes. Our decision making technique considers four modifiers as hard-constraints, which are *require*, *need*, *hate* and *don't accept*. Therefore, one might expect that options that do not satisfy preferences associated with *require* or *need*, or satisfy preferences associated with *hate* or *don't accept* have their rejection explained due to a cut-off value. Indeed, this intuition is the case, but we include other attributes in the cut-off group.

When users provide monadic preferences associated with a negative modifier, they indicate which values are not desired for certain attributes. Therefore, a rejected option has a cut-off attribute when its value satisfies a preference associated with a negative modifier, even if it is not considered a hard-constraint, that is, an undesirable value of an attribute option can be used as a reason to reject the option even though this preference is not a hard-constraint. Nevertheless, in some situations, options provide combinations of attribute values that make users, or an automated technique, to choose a certain option even though one or more of its attributes have an undesired value. Therefore, an option cannot have its rejection justified by a cut-off attribute if the chosen option also has the same undesired attribute value, or even worse. So, we first analyse the chosen option to verify the strongest negative modifier associated with its values, and then we explain rejected options by cut-off attributes when they either do not satisfy a hard-constraint or are associated with a negative modifier that is stronger than the modifiers associated with the chosen option.

To formally define this idea, we first define, in Equation 9-1, $min_{mod}(M_S)$, where $M_S \subseteq M$. This function returns the minimum modifier — i.e. the strongest negative, of a set of modifiers — and is based on the function f_m, which associates

modifiers with a real value, respecting the order established by the modifier scale (and in our decision technique we are currently using a logarithmic function).

$$min_{mod}(M_S) := m \mid m \in M_S \wedge \forall\, m'.(m' \in M_S \wedge m \neq m' \wedge f_m(m) \leq f_m(m')) \quad (9\text{-}1)$$

$min_{mod}(M_S)$ is used to capture the most negative modifier associated with an option, with a given lower bound (*don't need*), that is, if no negative modifier is associated with this option, the selected modifier is *accept*. Equation 9-2 defines the function $mostNegative(o)$, which makes this modifier selection.

$$\begin{aligned} mostNegative(o) := min_{mod}(\{\text{“}don't need''\} \cup \{mod \mid mod \in M \\ \wedge\, IsNegative(mod) \wedge \exists\, a.(PSM[o, a] = \langle empty, mod\rangle)\}) \end{aligned} \quad (9\text{-}2)$$

Finally, we define below when an attribute is considered a cut-off, following the informal description discussed previously.

Definition 9.2 *Let mod be a modifier from the set M, and $o_c \in Opt$ and $o_r \in Opt_r$. An attribute $a_{cutOff} \in Att$ said a **cut-off**, or $a_{cutOff} = CutOff(o_r, o_c)$, if we have:*

$$\begin{aligned} PSM[o_r, a_{cutOff}] = \langle \neg, \text{“}require''\rangle \vee PSM[o_r, a_{cutOff}] = \langle \neg, \text{“}need''\rangle \\ \vee\, (PSM[o_r, a_{cutOff}] = \langle empty, mod\rangle \wedge f_m(mod) < f_m(mostNegative(o_c))) \end{aligned}$$

If more than one attribute satisfy this property, we select most important one, i.e. if $w_{a_i}(o_r) < w_{a_j}(o_r)$, then the selected attribute is a_i.

The third pattern, namely Dominated Option, relies on the notion of domination, which is characterised by an option (dominant) that is better than another (dominated) with respect to one attribute, and at least as good with respect to the others. As this explanation does not involve any parameters, we do not discuss it in this section. So, next we address two patterns together: Minimum Requirements$^-$ and Minimum Requirements$^+$. In the scenario of these patterns, users have provided a set of constraints that lead to the elimination of options due to cut-off attributes, allowing the identification of a *consideration set*. In addition, the chosen option has no reason to be rejected, i.e. it satisfies all positive constraints and do not satisfy the negative ones, that is, for all attributes att we have $PSM[o_c, a] \neq \langle empty, IsNegative(modifier)\rangle$ and $PSM[o_c, a] \neq \langle \neg, IsPositive(modifier)\rangle$. If we have this scenario in the decision making process, and also there is one attribute that is decisive to choose one option from consideration set, which we refer to as *tie-breaker attribute*, we adopt these patterns to explain chosen and rejected options — excluding those rejected due to domination or cut-off attributes. The definition of the tie-breaker attribute is as follows.

Definition 9.3 *Let $a_{tieBreaker}$ and a be attributes from Att, and $o_c \in Opt$. $a_{tieBreaker}$ is said a **tie-breaker attribute**, or $TieBreaker(o_c)$, if there exists an option $o'_r \in Opt_r$*

rejected due to a cut-off value, i.e. $\exists\, a.(CutOff(o'_r, o_c) = a)$*, and for all the remaining rejected options* $o_r \in Opt_r$ *that* $\nexists a.(CutOff(o_r, o_c) = a)$*, we have* $OAPM[o_c, o_r, a_{tieBreaker}] = +$*. In addition, there is no* a' *that* $a' \neq a_{tieBreaker}$ *and* $OAPM[o_c, o_r, a_{tieBreaker}] = +$*, i.e.* $a_{tieBreaker}$ *is unique.*

We now proceed to the last two patterns, involving multiple attributes.

9.2.2 Multi-attribute Selection

In order to identify the *decisive criteria* to justify a decision made, required by the Decisive Criteria pattern, we first make definitions of concepts needed for identifying them. When two options are compared in our decision making technique, we identify pros and cons of these options with respect to each other. These are captured by the sets $Att^+(o_i, o_j)$ and $Att^-(o_i, o_j)$, which are sets of attributes that are pros and cons of o_i, respectively, and are defined as follows.

Definition 9.4 *Let* $o_i \in Opt$ *and* $o_j \in Opt$*. Then we define.*

$$Att^+(o_i, o_j) = \{a \mid a \in Att \wedge w_a(o_j) \times AttCost(o_j, o_i, a) > 0\}$$
$$Att^-(o_i, o_j) = \{a \mid a \in Att \wedge w_a(o_i) \times AttCost(o_i, o_j, a) > 0\}$$

As our technique identifies *how much* an option is preferred to another with respect to each attribute, we calculate the total pros and total cons of an option as shown below.

Definition 9.5 *Let* $o_i \in Opt$ *and* $o_j \in Opt$*. Then we define*

$$Pros(o_i, o_j) = \sum_{a^+ \in Att^+(o_i, o_j)} w_{a^+}(o_j) \times AttCost(o_j, o_i, a^+)$$
$$Cons(o_i, o_j) = \sum_{a^- \in Att^-(o_i, o_j)} w_{a^-}(o_i) \times AttCost(o_i, o_j, a^-)$$

The definition of decisive criteria is different for rejected and chosen options. The decisive criteria for rejecting an option consist of the subset of attributes whose values are enough for rejecting this option. For example, assume than an option X is chosen, and it has better values for the attributes a and b than an option Y has. In addition, $Cons(Y, X) > Pros(Y, X)$ — as we take into account trade-off contrast and extremeness aversion for making a choice, this may not hold. If we do not consider the benefit of X (cost of Y) with respect to b, and cons are still higher than pros, we can say that what matters is only the value of a of this option for

making the decision. This intuition is formalised below and, as more than one set of attributes may have this described characteristic, we also define a precedence for choosing one of these sets — we choose the simplest sets (in terms of the number of attributes), and among these, the strongest one (in terms of total pros).

Definition 9.6 *Let o_c be the option chosen from the set Opt by a decision making technique, and o_r a rejected option. The* ***decisive criteria*** *$D \subset Att^{-}(o_r, o_c)$ is the set of attributes such that $\sum_{a \in D} w_a(o_r) \times AttCost(o_r, o_c, a) < Pros(o_r, o_c)$. If there is an $S \subset Att^{-}(o_r, o_c)$ that also satisfies this property, and $S \neq D$, D is the decisive criteria if and only if*

$$(|D| < |S|) \vee (|D| = |S| \wedge \sum_{a \in D} w_a(o_r) \times AttCost(o_r, o_c, a) > \sum_{a \in S} w_a(o_r) \times AttCost(o_r, o_c, a))$$

Besides defining the decisive criteria for rejecting an option, it is essential to provide efficient means of identifying it, and this can be done by the execution of Algorithm 10. This algorithm includes attributes to the decisive criteria in a stepwise fashion, always including the attribute a_i whose value $w_{a_i}(o_r) \times AttCost(o_r, o_c, a_i)$ is the highest. At the moment that cons of the included attributes are higher than the pros, the algorithm stops, and returns the decisive criteria. If all attributes should be considered to make cons higher than pros, or if cons lower than pros (i.e. the rejection of o_r depends on the user-centric principles), there is no decisive criteria. In order to show that this proposed algorithm satisfies the conditions of Definition 9.6, we present the theorem below.

Theorem 9.7 *$DecisiveCriteria^{-}(o_r, o_c)$ returns the decisive criteria for rejecting o_r, or an empty set if this minimal set does not exist.*

Proof. We prove this theorem by contradiction. Assume that $D = DecisiveCriteria^{-}(o_r, o_c)$ and there is a subset $Out \subseteq D$, which is not part of the decisive criteria, and a subset $In \subseteq Att^{-}(o_r, o_c) \backslash D$, which is part of the decisive criteria. If $|In| > |Out|$, then $|In \cup (D \backslash Out)| > |D|$, and according to the Definition 9.6, $In \cup (D \backslash Out)$ is not the decisive criteria. If $|In| = |Out|$, as for all $attI \in In$ and $attO \in Out$, $w_{attI_i}(o_r) \times AttCost(o_r, o_c, attI_i) \leq w_{attO_i}(o_r) \times AttCost(o_r, o_c, attO_i)$ — as we sort $Att^{-}(o_r, o_c)$ — $\sum_{a \in In \cup (D \backslash Out)} w_a(o_r) \times AttCost(o_r, o_c, a) < \sum_{a \in D} w_a(o_c) \times AttCost(o_r, o_c, a)$, therefore contradicting the definition of decisive criteria. Finally, we analyse the case when $|In| < |Out|$. As the while loop ends at the first time that accumulated cons becomes higher than $Pros(o_r, o_c)$, and the last attribute added to D has the lower value for $w_a(o_r) \times AttCost(o_r, o_c, a)$, if we

remove any of the attributes of D, accumulated cons will become lower or equal to $Pros(o_r, o_c)$. So, $\sum_{a \in In \cup (D \setminus Out)} w_a(o_r) \times AttCost(o_r, o_c, a)$ has to be higher than $\sum_{a \in D \setminus \{x\}} w_a(o_r) \times AttCost(o_r, o_c, a)$, for all $x \in Out$; however, this is not possible, because we sort $Att^{-}(o_r, o_c)$, as shown in the previous case. And this completes the proof. ■

Algorithm 10: $DecisiveCriteria^{-}(o_r, o_c)$

Input: o_r: a rejected option; o_c: chosen option
Output: D: subset of Att containing the decisive criteria

```
1  SortedAtt⁻ ← Sort(Att⁻(o_r, o_c),
   a_i ≻ a_j ↔ w_{a_i}(o_r) × AttCost(o_r, o_c, a_i) > w_{a_j}(o_r) × AttCost(o_r, o_c, a_j));
2  AccumulatedCons ← 0;
3  D ← ∅;
4  while AccumulatedCons ≤ Pros(o_r, o_c) ∧ SortedAtt⁻ ≠ ∅ do
5      a ← Last(SortedAtt⁻);
6      SortedAtt⁻ ← SortedAtt⁻ − {a};
7      AccumulatedCons = AccumulatedCons + w_a(o_r) × AttCost(o_r, o_c, a);
8      D ← D ∪ {a};
9  if |D| < |Att⁻(o_r, o_c)| then
10     return D;
11 else
12     return ∅;
```

After showing how to identify the decisive criteria of rejected options, we analyse the case of the chosen option. The decisive criteria for justifying a chosen option can be either the set of attributes that the chosen option has the values better than at least half of the other options have, and no worse for the others; or (if this set does not exist), the decisive criteria for rejecting the option that has the lower pros and cons balance, when compared to the chosen option, which is seen as the "second best option." In both cases, we do not consider options rejected due to domination ($Expl(o, o_c) = \Psi_{dom}$) or cut-off attributes ($Expl(o, o_c) = \Psi_{cutOff}$). In order to identify the set of attributes of the first case, we define the concept of *best attributes* below.

Definition 9.8 *Let o_c be the option chosen from the set Opt by a decision making technique. The best attributes $B \subset Att$ is the set of attributes such that for all $a \in B$ and for all rejected options $o_r \in Opt_r*$ and $Opt_r* = Opt_r - \{o \mid Expl(o, o_c) = \Psi_{cutOff} \vee Expl(o, o_c) = \Psi_{dom}\}$, we have $OAPM[o_c, o_r, a] = +$, for at least $\frac{|Opt_r*|}{2}$ options, and $OAPM[o_c, o_r, a] = \sim$ for the remaining ones. Moreover, B is maximal in the sense of $\subset$.*

And now, we define the decisive criteria for the chosen option, which describes the two cases introduced above. Note that the decisive criteria for rejecting the option that has the lower pros and cons balance may not exist. This happens

because the lower pros and cons balance can be lower than 0, as we take into account trade-off contrast and extremeness aversion for making a choice.

Definition 9.9 *Let o_c be the option chosen from the set Opt by a decision making technique. The decisive criteria $D \subset Att$ is the best attributes B of o_c. If $B = \emptyset$, then D is the decisive criteria of an o_r, i.e. $DecisiveCriteria^{-}(o_r, o_c)$, such that $Pros(o_c, o_r) - Cons(o_c, o_r)$ is minimal, for all $o_r \in Opt_r$. Moreover, D exists if and only if $|D| \neq \emptyset$.*

The decisive criteria for a chosen option can be obtained by running Algorithm 11. The first part of the algorithm (lines 3–12) tries to identify the best attributes, and if they do not exist, then the second part (lines 14–15) tries to find the decisive criteria compared to the second best option.

Algorithm 11: $DecisiveCriteria^{+}(o_c)$

Input: o_c: chosen option
Output: D: subset of Att containing the decisive criteria

```
1  Opt_r* ← Opt − {o | o = o_c ∨ Expl(o, o_c) = Ψ_cutOff ∨ Expl(o, o_c) = Ψ_dom};
2  D ← ∅;
3  foreach a ∈ Att do
4      in ← true;
5      counter ← 0;
6      foreach o_r ∈ Opt_r* do
7          if OAPM[o_c, o_r, a] =~ then
8              counter ← counter + 1;
9          else if OAPM[o_c, o_r, a] ≠ + then
10             in ← false;
11     if in ∧ counter < |Opt_r*|/2 then
12         D ← D ∪ {a};
13 if D = ∅ then
14     o_r ← o | o ∈ Opt ∧ min(Pros(o_c, o) − Cons(o_c, o));
15     D ← DecisiveCriteria^−(o_r, o_c);
16 return D;
```

A chosen or rejected option may not be associated with a set of attributes, which are the decisive criteria for making the decision in different cases, and therefore the **Decisive Criteria** pattern cannot be applied, so the last explanation pattern — **Trade-off Resolution**— has to be adopted to justify the choice for the user. We next describe these cases for the chosen option, and then later rejected options.

For explaining a chosen option, which does not have a set of attributes that are the decisive criteria of the decision, we have three cases to analyse, which consist of the reasons why there is no decisive criteria. First, a chosen option o_c may not have one or more attributes that are better than the attributes of all other options, and also the pros and cons balance of second best option may

be negative, that is, $Pros(o_c, o_r) < Cons(o_c, o_r)$ — meaning that the trade-off contrast and/or extremeness aversion are responsible for choosing o_c instead of o_r. For explaining this scenario, we have two alternatives, which depend on the existence of a set $D \subset Att$, which $D = DecisiveCriteria^{-}(o_c, o_r)$. When D exists, the provided explanation highlights that o_r has D pros (i.e. "*even though* o_r *is better considering* a_x, a_y, *etc.*"), and states that o_c has a better cost-benefit relationship, as $ToContrast(o_r, o_c) > 0 \vee ExtAversion(o_r, o_c) > 0$. When these decisive criteria do not exist, we have a procedure to select both decisive pros and decisive cons, shown in Algorithm 12, which identifies the maximal set of pros that should be considered for enabling the existence of a decisive criteria for rejecting o_c. Therefore, $DecisiveProsCons(o_c, o_r)$, for an o_r whose pros are higher than cons when compared to the chosen option, identifies the cons that should be shown in the "*even though*" part of the explanation, and also the pros that should be mentioned, which compensate cons. Moreover, the cost-benefit relationship is also highlighted as the trade-off contrast and extremeness aversion play an important role in the decision.

Algorithm 12: $DecisiveProsCons(o_i, o_j)$

Input: $o_i, o_j \in Opt$

Output: $\langle P, C \rangle$: subsets of Att, which represents pros and cons of o_i

1 $SortedAtt^{+} \leftarrow$ `Sort(`$Att^{+}(o_i, o_j)$, $a_i \succ a_j \leftrightarrow w_{a_i}(o_j) \times AttCost(o_j, o_i, a_i) > w_{a_j}(o_j) \times AttCost(o_j, o_i, a_j)$`)`;

2 $RemainingPros \leftarrow Pros(o_i, o_j)$;

3 $P \leftarrow \emptyset$;

4 $C \leftarrow \emptyset$;

5 **while** $C = \emptyset \wedge SortedAtt^{+} \neq \emptyset$ **do**

6 $a \leftarrow$ `Last(`$SortedAtt^{+}$`)`;

7 $SortedAtt^{+} \leftarrow SortedAtt^{+} - \{a\}$;

8 $RemainingPros = RemainingPros - w_a(o_j) \times AttCost(o_j, o_i, a)$;

9 $P \leftarrow P \cup \{a\}$;

10 $C \leftarrow DecisiveCriteria^{-}(o_i, o_j, RemainingPros)$;

// $DecisiveCriteria^{-}(o_i, o_j, RemainingPros)$ `above is` $DecisiveCriteria^{-}(o_i, o_j)$ `but considering only the remaining pros`

11 **if** $C = \emptyset$ **then**

12 $C \leftarrow Att^{-}(o_i, o_j)$;

13 **return** $\langle P, C \rangle$;

In case o_c has the best pros and cons balance, but none of the attributes have the best values in comparison with other acceptable options (i.e. the ones not excluded due to a cut-off value or domination), we use the second best option — the option o_r that has the minimum pros and cons balance ($Pros(o_c, o_r) - Cons(o_r, o_c)$) — to explain the decision. This scenario is explained by finding the decisive criteria for rejecting the second best option, but this case was already covered in previous section. Therefore, there is only one case left, that is, o_c has the best pros and cons

balance, but there is no decisive criteria to choose it over the second best option. The explanation given in this case is based on the same algorithm adopted before, but used in the opposite direction — $DecisiveProsCons(o_r, o_c)$ — we identify key attributes of the second best option, which are removed so that we can identify decisive criteria, and the explanation states that even though o_r (the second best option) has better values associated with the key attributes (o_c's disadvantages), the values of the attributes that are the decisive criteria compensate these disadvantages.

Rejected options may also not have associated decisive criteria, since all attributes that characterise the cons of a rejected option may play a role in the decision between this option and the chosen option, or the trade-off contrast and extremeness aversion may have played a crucial role in the decision. So, to justify rejected options in these situations, the reasoning to build an explanation is similar to that made for explaining the chosen option. First, we analyse if the rejected option o_r has a better pros and cons balance than the chosen option $Pros(o_r, o_c) > Cons(o_r, o_c)$. If so, we adopt the same approach used previously. (a) If there is a set of attributes that characterises the decisive criteria for choosing o_r instead of o_c, i.e. $DecisiveCriteria^{-}(o_c, o_r)$, we highlight these positive aspects of o_r, and state that, nevertheless, o_r has a worse cost-benefit relationship when compared to o_c. (b) If there is no decisive criteria, we select the decisive pros and cons $\langle P, C\rangle = DecisiveProsCons(o_c, o_r)$, and besides mentioning only the cost-benefit relationship of o_c, we also highlight its decisive pros. This procedure is also the one applied when $Pros(o_r, o_c) \leq Cons(o_c, o_r)$, but there is no decisive criteria to justify the decision.

We have shown different ways of explaining the trade-off resolution in order to justify a chosen or a rejected option. In Table 9.1 we summarise these different scenarios and, as explanations that follow the Trade-off Resolution Pattern receive as parameters pros and cons to be made explicit, we show how they are obtained. In some scenarios, a constant argument — better or worse cost-benefit relationship — may be part of the explanation.

9.3 Choosing and Generating an Explanation

After showing how parameters are selected to be part of explanations, we now present how we choose an explanation to be given. First, we introduce the representation of each explanation type in Table 9.2. This representation is the information that we need for generating an explanation according to the templates proposed in our explanation patterns. We extend these patterns by including Domination as an explanation of a chosen option, which is applied when the chosen option dominates all the other ones. This is not reported as a pattern, as this situation

Chosen Option			
$\exists\, o_r.(Pros(o_c, o_r) < Cons(o_c, o_r))$?	**Decisive Criteria**	**Pros**	**Cons**
Yes	$DecisiveCriteria^{-}(o_c, o_r) \neq \emptyset$	Cost-benefit relationship	$DecisiveCriteria^{-}(o_c, o_r)$
Yes	$DecisiveCriteria^{-}(o_c, o_r) = \emptyset$	P returned by $DecisiveProsCons(o_c, o_r)$ Cost-benefit relationship	C returned by $DecisiveProsCons(o_c, o_r)$
No	$DecisiveCriteria^{-}(o_{2^{nd}Best}, o_c) \neq \emptyset$	Decisive Criteria pattern	
No	$DecisiveCriteria^{-}(o_{2^{nd}Best}, o_c) = \emptyset$	C returned by $DecisiveProsCons(o_{2^{nd}Best}, o_c)$	P returned by $DecisiveProsCons(o_{2^{nd}Best}, o_c)$
Rejected Options			
$Pros(o_r, o_c) > Cons(o_r, o_c)$?	**Decisive Criteria**	**Pros**	**Cons**
Yes	$DecisiveCriteria^{-}(o_c, o_r) \neq \emptyset$	$DecisiveCriteria^{-}(o_c, o_r)$	Cost-benefit relationship
Yes	$DecisiveCriteria^{-}(o_c, o_r) = \emptyset$	C returned by $DecisiveProsCons(o_c, o_r)$	P returned by $DecisiveProsCons(o_c, o_r)$ Cost-benefit relationship
No	$DecisiveCriteria^{-}(o_r, o_c) \neq \emptyset$	Decisive Criteria pattern	
No	$DecisiveCriteria^{-}(o_r, o_c) = \emptyset$	P returned by $DecisiveProsCons(o_r, o_c)$	C returned by $DecisiveProsCons(o_r, o_c)$

Table 9.1: Trade-off explanations: selection of pros and cons to be shown.

Explanation Type	Representation	Parameters
Critical Attribute	$\Psi_{crit}(o_c, a)$	$o_c \in Opt$ $a \in Att \wedge att = CriticalAttribute(o_c)$
Domination	$\Psi_{dom^+}(o_c)$ or $\Psi_{dom^-}(o_r, o_c)$	$o_c, o_r \in Opt$
Cut-off	$\Psi_{cutOff}(o_r, a)$	$o_r \in Opt$ $a \in Att \wedge a = CutOff(o_r, o_c)$
Minimum Requirements$^+$	$\Psi_{minReq^+}(o_c, a)$	$o_c \in Opt$ $a \in Att \wedge a = TieBreaker(o_c)$
Minimum Requirements$^-$	$\Psi_{minReq^-}(o_r, o_c, a)$	$o_c, o_r \in Opt$ $a \in Att \wedge a = TieBreaker(o_c)$
Decisive Criteria	$\Psi_{decisive}(o, target, atts)$	$o \in Opt$ $target \in \{chosen, rejected\}$ $atts \subset Att$
Trade-off Resolution	$\Psi_{tradeOff}(o, target, atts_P, atts_C, cb)$	$o \in Opt$ $target \in \{chosen, rejected\}$ $atts_P, atts_C \subset Att$ Pros and Cons $cb \in \{true, false\}$ Cost-benefit relationship is an argument?

Table 9.2: Explanation Types.

is very unlikely to occur but, as it is possible, we take it into consideration.

Explanations presented in Table 9.2 are all possible explanations that can be given either to justify choosing an option or rejecting an option. In some situations, more than one explanation can be given for justifying a decision, but we choose one of them based on a precedence order, which is shown below for the chosen option.

$$\Psi_{crit} \triangleright \Psi_{dom} \triangleright \Psi_{minReq^+} \triangleright \Psi_{decisive} \triangleright \Psi_{tradeOff} \tag{9-3}$$

In order to select an explanation according to this order, we propose Algorithm 13 that makes this selection. In this algorithm, we use $dominates(o_i, o_j)$ presented in Definition 6.6.1 (Chapter 6). The remaining procedures or functions were introduced in this chapter.

Similarly, for rejected options, we also establish a precedence order for the possible explanation types, as presented below. Algorithm 14 describes how an explanation is selected for a particular rejected option. As in the previous algorithm, the main idea is to verify if the conditions for using a pattern are satisfied, following the precedence order.

$$\Psi_{crit} \triangleright \Psi_{cutOff} \triangleright \Psi_{dom} \triangleright \Psi_{minReq^-} \triangleright \Psi_{decisive} \triangleright \Psi_{tradeOff} \tag{9-4}$$

9.4 The Apartment Example: Illustrating our Approach

Now, we come back to our running example introduced in Chapter 6 to show how explanations to justify the choice for the apartment Ap_B are generated. As there is neither an attribute that is a critical attribute for the decision, nor one that is a tie-breaker, the Critical Attribute and the Minimum Requirements patterns cannot be applied. Next, we discuss each option individually and provide for them an explanation for their rejection, or choice, in case of Ap_B.

Algorithm 13: $Explanation(o_c)$

Input: $o_c \in Opt$: chosen option
Output: Ψ: explanation to justify the choice

if $\exists\, a.(CriticalAttribute(o_c) = a)$ **then**
  return $\Psi_{crit}(o_c, CriticalAttribute(o_c))$;
if $\forall\, o_r.(dominates(o_c, o_r))$ **then**
  return $\Psi_{dom}(o_c)$);
$ok \leftarrow true$;
foreach $a \in Att$ **do**
  $\langle x, mod \rangle \leftarrow PSM[o_c, a]$;
  if $(x = empty \wedge IsNegative(mod)) \vee (x = \neg \wedge IsPositive(mod))$ **then**
    $ok \leftarrow false$;
if $ok \wedge \exists\, a.(TieBreaker(o_c) = a)$ **then**
  return $\Psi_{minReq^+}(o_c, TieBreaker(o_c)))$;
$D \leftarrow DecisiveCriteria^+(o_c)$;
if $D \neq \emptyset$ **then**
  return $\Psi_{decisive}(o_c, accept, D)$;
else
  if $\exists\, o_r.(Pros(o_c, o_r) < Cons(o_c, o_r))$ **then**
    $D \leftarrow DecisiveCriteria^-(o_c, o_r)$;
    if $D \neq \emptyset$ **then**
      return $\Psi_{tradeOff}(o_c, accept, \emptyset, D, true)$;
    else
      $\langle P, C \rangle \leftarrow DecisiveProsCons(o_c, o_r)$;
      return $\Psi_{tradeOff}(o_c, accept, P, C, true)$;
  else
    $o_{2^{nd}Best} \leftarrow o \mid o \in Opt \wedge min(Pros(o_c, o) - Cons(o_c, o))$;
    $\langle P, C \rangle \leftarrow DecisiveProsCons(o_{2^{nd}Best}, o_c)$;
    return $\Psi_{tradeOff}(o_c, accept, C, P, false)$;

Algorithm 14: $Explanation(o_r, o_c)$

Input: $o_r \in Opt$: a rejected option; $o_c \in Opt$: chosen option
Output: Ψ: explanation to justify the choice

if $\exists\, a.(CriticalAttribute(o_c) = a)$ **then**
  return $\Psi_{crit}(o_c, CriticalAttribute(o_c))$;
if $dominates(o_c, o_r))$ **then**
  return $\Psi_{dom}(o_r, o_c))$;
if $\exists\, a.(CutOff(o_r, o_c) = a)$ **then**
  return $\Psi_{cutOff}(o_r, CutOff(o_r, o_c))$;
if $Explanation(o_c)$ *is-a* Ψ_{minReq^+} **then**
  return $\Psi_{minReq^-}(o_r, TieBreaker(o_c)))$;
$D \leftarrow DecisiveCriteria^-(o_r, o_c)$;
if $D \neq \emptyset$ **then**
  return $\Psi_{decisive}(o_r, reject, D)$;
else
  if $Pros(o_r, o_c) > Cons(o_r, o_c)$ **then**
    $D \leftarrow DecisiveCriteria^-(o_c, o_r)$;
    if $D \neq \emptyset$ **then**
      return $\Psi_{tradeOff}(o_r, reject, D, \emptyset, true)$;
    else
      $\langle P, C \rangle \leftarrow DecisiveProsCons(o_c, o_r)$;
      return $\Psi_{tradeOff}(o_r, reject, C, P, true)$;
  else
    $\langle P, C \rangle \leftarrow DecisiveProsCons(o_r, o_c)$;
    return $\Psi_{tradeOff}(o_r, reject, P, C, false)$;

Ap_A**.** Option Ap_A is neither dominated by Ap_B nor has a value that makes it be cut-off. As the $DecisiveCriteria^{-}(Ap\_A, Ap\_B) \neq \emptyset$, rejecting this option is explained by stating the criteria that were decisive for its rejection. The explanation is $\Psi_{decisive}(Ap\_A, reject, \{zone, uni\})$, written as follows — we use the full attribute names, i.e. instead of "*uni*," we use "*distance from the university*."

> *Option* ***Ap_A*** *was rejected because of its* ***zone*** *and* ***distance from the university****.*

Ap_B**.** Option Ap_B is the chosen option, and, as mentioned before, there is no critical or tie-breaker attribute to justify the decision, and Ap_B also does not dominate all the remaining options. By executing Algorithm 13 the explanation type returned is decisive criteria, as $DecisiveCriteria^{+}(Ap\_B) = \{zone\}$, and therefore the explanation is $\Psi_{decisive}(Ap\_B, accept, \{zone\})$. Based on this returned explanation, we are able to generate the following statement, according to the pattern template.

> *Option* ***Ap_B*** *was chosen because of its* ***zone****.*

Ap_C**.** Option Ap_C has many attributes that are better than other options have, but its attribute $zone$ has a value that is not acceptable for the user. So, $CutOff(Ap\_C, Ap\_B) = zone$, and the given explanation is $\Psi_{cutOff}(Ap\_C, zone)$, informally written as shown below.

> *Option* ***Ap_C*** *was rejected because it does not satisfy constraints associated with* ***zone****.*

Ap_D**.** Even though option Ap_D is dominated by Ap_A, domination is not used as an explanation because Ap_A is not the chosen option. As $DecisiveCriteria^{-}(Ap\_D, Ap\_B) = \{uni\}$, the decisive criteria pattern is then used.

> *Option* ***Ap_D*** *was rejected because of its* ***distance from the university****.*

Ap_E**.** As Ap_E is not dominated by Ap_B, has acceptable attributes values and has no decisive criteria to justify its rejection, we have to identify the pros and cons that support the choice for Ap_B. $Pros(Ap\_E, Ap\_B) < Cons(Ap\_E, Ap\_B)$, but $DecisiveCriteria^{-}(Ap\_E, Ap\_B) = \emptyset$, so we have to execute $DecisiveProsCons(Ap\_E, Ap\_B)$. As a result, we have $\langle\{brand\}, \{uni\}\rangle$, and thus the explanation is $\Psi_{tradeOff}(Ap\_E, reject, \{brand\}, \{uni\}, false)$.

> *Even though option **Ap_E** provides better **brand** than the chosen option, it has worse **distance from the university**.*

Ap_F**.** The explanation for Ap_F is similar to that for Ap_E, but in this case $Pros(Ap\_F, Ap\_B) > Cons(Ap\_F, Ap\_B)$, indicating that the user-centric principles played an essential role in the decision. So executing $DecisiveProsCons(Ap\_B, Ap\_F)$, we have as result $\langle\{price\}, \{brand\}\rangle$, and thus the explanation is $\Psi_{tradeOff}(Ap\_F, reject, \{brand\}, \{price\}, true)$.

> *Even though option **Ap_F** provides better **brand** than the chosen option, it has worse **price** and **cost-benefit relationship**.*

9.5 Comparison with Related Work and Performance Evaluation

We now will discuss a comparison of our approach with two existing approaches (Klein and Shortliffe 1994, Labreuche 2011), which also address the selection of attributes to be part of the explanations. These approaches focus on explaining why one particular option is better than another, and not the whole explanation — they mainly focus on the attribute selection process. As many explanations are similar for the different options, we show the explanation generated for options Ap_E and Ap_F.

Both approaches assume that the decision is made based on MAUT, and the value of an option (how much an option is preferred) is a real number between 0 and 1, calculated by the weighted sum of values (how much an attribute is preferred) of attributes. We use our $Cost(o_i, o_j)$ function to select attributes using these approaches. Klein and Shortliffe's approach (Klein and Shortliffe 1994) relies on the concept of compellingness, which is used to select attributes whose values are above a threshold (calculated based on the average and standard deviation of values associated with option attributes). Labreuche's approach (Labreuche 2011), on the other hand, has different strategies for selecting attributes. Table 9.3 shows which attributes are selected for each of these approaches.

It can be seen that these two approaches and ours differ on the selected decisive criteria. Klein and Shortliffe's approach selects attributes based on a threshold, which in some cases does no capture all attributes needed to support the decision or selects too many. Therefore, as Labreuche argued, there is no formal reason why an attribute should be selected. Labreuche addressed this limitation using another approach — he analyses attributes weights (comparing them with average weights or switching them). As a consequence, in some scenarios, attributes that are not important (by having very low weight) and are associated with small

Approach	Ap_E	Ap_F
Klein and Shortliffe's approach	The attribute **distance from the university** provides the most compelling reason to prefer **Ap_B** over **Ap_E**.	**Ap_B** is preferred to **Ap_E** since criterion **zone** for which **Ap_B** is better than **Ap_E** is more important than criterion **brand** for which **Ap_B** is worse than **Ap_E**.
Labreuche's approach	The attributes **distance from the university**, **brand**, and **stars** are reasons to prefer **Ap_F** over **Ap_B**. The attributes **distance from the station**, and **price** are reasons to prefer **Ap_B** over **Ap_F**.	**Ap_B** is preferred to **Ap_F** since the intensity of preference **Ap_B** over **Ap_F** on **distance from the station**, and **price** is significantly larger than the intensity of preference of **Ap_F** over **Ap_B** on **distance from the university**, **brand**, and **stars**.

Table 9.3: Comparison of selected decisive criteria.

values are selected as part of explanations, and they may be irrelevant for the decision. Our approach follows Klein and Shortliffe's idea that the combination of attribute weights and values (in our case costs) are both important for selecting the decisive criteria, but we give a formal reason why an attribute should be selected as part of the explanation. It can be seen that both approaches selected all pros and cons to explain option Ap_F, which occurred because there is no compelling attribute according to the first approach, and only the remaining case anchor of the second approach applies to this option. This situation is more likely to happen when the user-centric principles have an important role in the decision.

After making this comparison with related work, we present a performance evaluation of our approach. We have implemented the proposed algorithms, and used this implementation to evaluate our approach and also the distribution of explanation types in a real scenario. This evaluation is based on the study of how humans express preferences in which we have preferences written in natural language by people, which were expressed with our preference language. These preference specifications (113 in total) were used to evaluate our decision making technique. We now use them to produce explanations based on the reasoning traces generated during the execution of the decision making technique.

On average, our technique takes $2.12ms$ (on a Intel Core 2 Quad $2.66GHz$, $4GB$ of RAM) to generate explanations, standard deviation 1.85, minimum $0.0ms$ (actually, $< 0.0ms$), and maximum $17ms$. This indicates that our explanation technique performs well in a real scenario: 144 options, 61 attributes, and realistic sets of preferences provided by people. All algorithms are executed at maximum in $6ms$, and sometimes less than $1ms$, as reported in Table 9.4, which also shows the percentage of each generated explanation types (distribution column). The *Explanation Generation Time* column reports the times to execute the algorithms

Explanation Type	Distribution	Explanation Generation Time (ms)				Pattern time (ms)		
	%	Min	Max	Avg	StDev	Pos	Neg	All
Chosen Option								
Critical Attribute	0.00%						0.03448	0.03448
Domination	1.72%	0.00	0.00	0.00000	0.00000			
Minimum Requirements	5.17%	0.00	1.00	0.83333	0.37268			
Decisive Criteria	58.62%	0.00	6.00	0.75000	0.81123	0.05882	0.30000	0.14815
Trade-off Resolution	21.55%	0.00	3.00	1.08000	0.56000			
Random	12.93%	1.00	2.00	1.06667	0.24944			
Total	100.00%	0.00	6.00	0.85345	0.71020			
Rejected Options								
Critical Attribute	0.00%							
Cut-off	13.03%	0.00	1.00	0.00117	0.03420	0.00000	0.00000	0.00000
Domination	42.78%	0.00	1.00	0.00383	0.06178	0.00014	0.00176	0.00097
Minimum Requirements	0.44%	0.00	1.00	0.01389	0.11703			
Decisive Criteria	31.64%	0.00	1.00	0.02188	0.14628	0.01409	0.00942	0.01285
Trade-off Resolution	11.06%	0.00	3.00	0.05656	0.23803	0.03065	0.00524	0.02803
Random	1.07%	0.00	1.00	0.01705	0.12944			
Total	100.00%	0.00	3.00	0.01420	0.11975			

Table 9.4: Explanation Evaluation.

$Explanation(o_c)$ and $Explanation(o_r, o_c)$, which generate explanations for the chosen and rejected options, respectively, while the *Pattern time* column shows the times to execute the algorithm used to generate an explanation of a particular pattern, such as the execution of the algorithm $DecisiveCriteria^{+}(o_c)$ to generate the explanation for the chosen option according to the Decisive Criteria pattern.

Besides the explanation types presented in this chapter, there is an additional one: *random*. This explanation type was added to address a scenario not discussed before, which consists of having options with the exact same values for attributes. Therefore, in this case we select one of them randomly. Note that the random choice is made between two or three options with the exact same values, and not among the whole set of available options.

The difference in pattern execution times is mainly because the first explanation generated is for the chosen option, but to verify the applicability of the Domination and Minimum Requirements^{+} patterns, we must analyse dominated options and cut-off attributes. As this information is stored when first evaluated, it does not need to be evaluated again to obtain explanations for rejected options.

9.6 Final Considerations

In this chapter, we presented a technique to generate explanations for users to justify choices made by our decision making technique. The technique is based on proposed guidelines and patterns, and provides a means of identifying parameters of explanation templates, which are part of the patterns. Our technique not only identifies these parameters, but also provides an algorithm to choose which explanation should be used in different cases. We illustrated the explanation generation with an apartment example, showing how we justify a chosen option,

and the rejection of the remaining ones. Moreover, we presented a performance evaluation of our approach, showing its efficiency, and the distribution of explanations produced in a real scenario. As this evaluation does not covers user acceptance of our explanation generation technique, we present in next chapter a user study performed to evaluate not only this technique, but also our preference metamodel and decision technique.