Masterarbeit

in

Kognitive Systeme

zum

Thema

Analysing Visual Appearance by
Featural And Relational Similarity

- How Different Are Smartphones?

im Studiengang Angewandte Informatik
der Fakultät Wirtschaftsinformatik und
Angewandte Informatik
der Otto-Friedrich Universität Bamberg

don

Markus Wachter

11.07.2012

Gutachter: Prof. Ute Schmid
# Table of Contents

1. Introduction ........................................................................................................... 3  
2. Similarity .................................................................................................................. 4  
3. Smartphone Datastructure ....................................................................................... 4  
   3.1 Numerical Features .............................................................................................. 5  
   3.2 Numerical Relations ............................................................................................ 5  
   3.3 Design Features ................................................................................................... 5  
   3.4 Categorical Design Features .............................................................................. 6  
   3.5 Filter Features .................................................................................................... 8  
4. Similarity Calculation ............................................................................................... 10  
   4.1 Similarity of Numerical Features And Relations ............................................... 10  
   4.2 Similarity of Categorical Design Features ......................................................... 11  
   4.3 Smartphone Similarity ....................................................................................... 12  
   4.4 Weighted Similarity Calculation ........................................................................ 11  
   4.5 Filtered Similarity Calculation .......................................................................... 12  
5. Smartphone Similarity Tool .................................................................................... 13  
   5.1 Requirements and Smartphone Data ................................................................. 13  
   5.2 Searching similar Smartphones ....................................................................... 13  
   5.3 Searching a new Smartphone .......................................................................... 14  
   5.4 Feature Battle – Smartphone Battle ................................................................. 14  
6. Conclusion ............................................................................................................. 16  
   References ............................................................................................................... 18  
A. Appendix ............................................................................................................... 19  
   A1 Feature: hardkey ................................................................................................. 19  
   A2 Feature: curveCorner ......................................................................................... 20  
   A3 Feature: curveEdge ............................................................................................. 22  
   A4 Smartphone XML Example Data ....................................................................... 23  
   A5 CD with the Smartphone Similarity Tool ......................................................... 24
1 Introduction

In the last ten to fifteen years mobile phones are our daily companion and no one can now imagine, how to live a life without it. At the beginning of the mobile age, phones were big, heavy and had only a limited performance but if you don’t know better, you wouldn’t mind. Over the last few years the performance and the design has changed dramatically. The modern ‘smartphones’ as they are called today, have more power and technical features than my notebook which I bought just six years ago. But not only the performance and the technical features have changed, but also the appearance and the design. I would say, that Steve Jobs was the first pioneer that changed or rather started a new smartphone age with the first smartphone which you can handle and use only by touching the screen. Before that, you used hardkeys or a whole keyboard to use and control the phone. With the first iPhone from Apple this using behavior changed and now you control your phone respectively smartphone with the whole screen, the touch screen. So now there was some space available on the phone you needed for the keyboard before. The consequence was of course, to make the screen bigger so that the smartphone almost consists of a big touch screen. Of course other manufacturers picked up this idea and also build their phones with bigger screens and more technical features like the first iPhone from Apple. No wonder that now almost every smartphone looks the same, with a big screen, big length and width of the phone to show as more information as they can and of course the thinner the better for a comfortable feeling if you carry around and hold it in your hand.

The technical features are almost the same in every smartphone, like a good processor (now there are quad-core CPUs), big memory and almost every connectivity feature available (WLAN, Bluetooth, etc.). So with all this technical features in the smartphone, you only can distinguish the devices from each other by the specific manufacturer design they use to build their phones. Some have rough edges, some are more rounded than others, some are bigger, some smaller. The design now is the only way to sometimes distinguish between two smartphones. An example for building devices that look almost the same, is the current patent conflict between Apple and Samsung. Apple blames Samsung for stealing the idea respectively to infringe the patent of Apple, which they use to build and design their iPad.

In this thesis the visual appearance of smartphones will be analyzed by calculating a similarity value between two devices. This similarity calculation is affected by featural and relational parameters which will describe the design and shape of the smartphones. In section 2 a short overview of how similarity can be calculated is given and what features you can retrieve if you represent a smartphone. In section 3 a possible representation of a smartphone using XML Schema is introduced. Section 4 deals with the similarity calculation which is used for the developed SmartphoneSimilarity-Tool described in section 5.
2 Similarity

Analysing the similarity of two objects is not easy and depends on which information is used to calculate the similarity. Different types of similarity measures are applicable. For instance, similarity between metrical or **numeric features**, such as the size (e.g. length, width, height) of an object is typically calculated as Euclidian distance between two objects. Similarity of **categorical features** can be measured with the contrast model (Goldstone & Son, 2004).

There is a variety of information when classifying an object. Considering a modern smartphone, a representation of this object might contain the following characteristics:

- **numeric visual features** such as the length or display diagonal of a smartphone
- **numeric non-visual features** such as the weight of the smartphone
- **numeric visual relations** such as the proportion of the length to the width or better the width to the height, which could be a haptic feature that determines how comfortable the phone lays in your hand and can be used
- **categorical design features** such as the shaped edges and shaped curves of a smartphone (ordinal values)
- **categorical visual features** such as the operating system like Android or Windows Phone
- **categorical non-visual features** such as the manufacturer
- **qualitative numeric feature** such as the price of a smartphone.

Looking at these characteristics, it could be very difficult to consider all of these features to calculate a similarity between two smartphones. Furthermore, some of the features might be more relevant to someone than other features. Considering this, the similarity of one feature should be weighted more than the similarity of another feature. Before we see how the similarities of the features are calculated, we consider a representation of the smartphone.

So let’s see how and in which kind of datastructure we can represent a smartphone and programmatically handle these data. In the next section a representation of the smartphone is given with a XML Schema definition and we see how the numerical and categorical features are represented.

3 Smartphone Datastructure

To represent a smartphone with all its characteristics an XML datastructure is used to describe the smartphone. To make sure that every phone has the same structure, an XML Schema Definition (XSD) is needed. In the following we see a schema for a smartphone, which was used in the developed application.
3.1 Numerical Features

A smartphone XML element contains elements that describe the numerical features like `sizeLength`, `sizeWidth`, `sizeHeight` the `weight` and the `displayDimension`, which will be numeric double values (e.g. `length=115` or `height=12.3`).

```xml
<xsd:element name="sizeLength" type="xsd:double" />
<xsd:element name="sizeWidth" type="xsd:double" />
<xsd:element name="sizeHeight" type="xsd:double" />
<xsd:element name="weight" type="xsd:double" />
<xsd:element name="displayDimension" type="xsd:double" />
<xsd:element name="price" type="xsd:int" />
```

The price of the smartphone is also a numerical feature but not for the visual similarity relevant. This parameter will only provide a filter search for the developed application (see section 5).

3.2 Numerical Relations

The numerical features `sizeLength`, `sizeWidth` and `sizeHeight` provide also a numerical relation for the smartphone. Three numerical relations can be extracted from the numerical size values. These numerical relations are:

- relation between `sizeLength` <> `sizeWidth`
- relation between `sizeWidth` <> `sizeHeight`
- relation between `sizeLength` <> `sizeHeight`

Consider the second relation `sizeWidth` <> `sizeHeight`, we can say, that this relation is a measure of how good you can hold the smartphone with one hand, because if the phone is really wide and thick, you probably have problems holding the phone. It is not a coincidence that manufacturers like Apple, HTC or Samsung etc. produce smartphones that are only a few millimeters high by getting wider and have a bigger display diagonal. These relations could also be important for the visual appearance and visual similarity of the smartphones.

Considering the numerical features we now can easily calculate a similarity between two smartphones (see section 4.1). But not only numeric features are relevant for the similarity of two smartphones.

3.3 Design Features

Another design feature like a hardkey on the front side of the smartphone is also relevant. In my point of view, a smartphone with a large display (as almost every current smartphone), does not need a so called hardkey on the front view of the phone. This type of key is a physical key that you have to press to use its functionality. The opposite of this hardkey are softkeys, which you do not press, but slightly touch on the big touch screen. Obviously this key is another design parameter that would distinguish one smartphone to another and is not preferred by
everyone. We represent this key as a Boolean value. Consider figure 1 for the two values of this feature.

\[
\text{<xsd:element name="hardkeys" type="xsd:boolean" />}
\]

3.4 Categorical Design Features

Now let’s see the categorical design features like the shaped edges and shaped curves of the smartphone. Every smartphone has one value of this ordinal characteristic. The feature \textit{curveCorner} and \textit{curveEdge} are categorical represented and describe the shape of the four corners (\textit{curveCorner}) of the smartphone and the shape of the edges (\textit{curveEdge}) when you look on the top side of the phone. Consider figure 3 with the four values of the category \textit{curveCorner}. The four values have the ordinal order (from an edged curve to a round curve):

\textit{curveCorner}:  
- EDGED
- EDGED_LOW_ROUNDING
- STRONG_ROUNDING
- GLOBAL_ROUNDING

The feature \textit{curveCorner} is more a visual characteristic than the feature \textit{curveEdge}, because you could say it is not only a visual but more a haptical feature as it describes the shape of the edges of the left and right side of the smartphone. These edges decide how the phone is formed and how comfortable it lays in your hand. For someone who uses the phone more with only just one hand, it could be more comfortable if the shape of the edges is more like the shape of the holding hand and so has the \textit{curveEdge} value STRONG_ROUNDEING_LONGER_FLATTENING (figure 4). The ordinal order is as follows (from rough edges to rounder edges with a flattening shape on the back of the smartphone):
**curveEdge:**
- EDGED_NO_ROUNDING_NO_FLATTENING
- EDGED_ROUNDING_NO_FLATTENING
- EDGED_ROUNDING_LOW_FLATTENING
- STRONG_ROUNDING_LONGER_FLATTENING

![Figure 3: curveCorner with four different values from an edged curve to a round curve](image-url)
3.5 Filter Features

Filter features are no numeric or categorical features we can calculate with. A filter feature could be the manufacturer of a phone. Let’s say someone only prefers smartphones from a specific manufacturer, we should consider this in a preferred filtered similarity calculation (see in section 4.5). This kind of filter was also implemented in the application (see section 5).

Another characteristic of a smartphone is the operating system. This property is not actually a feature you can measure and calculate a similarity value. It only provides a filter for the similarity regarding a specific operating system. You could now ask, why even use the operating system as a filter for visual similarity? The appearance of a smartphone is not only determined by the design parameters (the sizes and the design of the curves and edges), but also by the visual appearance of the display content. Compared to a personal computer, someone may not like a Mac PC because it is from Apple, but more because the operating system on the PC is not like Windows, or if you consider a Linux operating system, which is much more different than a Microsoft Windows operating system. The operating system on a phone like Android (as on Samsung or HTC), iOS (Apple) or Windows Phone also differ from each other and maybe you won’t like such kind of operating system on your phone.

Compare the three different operating systems in figure 5. Notice the three different types of Android operating systems, because on the HTC Desire HD (a) there is the manufacturers own HTC-Sense user interface (UI), where on the
Samsung Galaxy S II (b) there is Samsungs own TouchWiz UI and on the Samsung Galaxy Nexus (c) there is the normal Android operating system, (d) is the operating system iOS from Apple on the Apple iPhone 4, (e) is Windows Phone from Microsoft on the Nokia Lumina 800 and (f) is the Symbian operating system on the Nokia 500.

Figure 5: The different operating systems:
(a) HTC Desire HD with Android and HTC-Sense UI
(b) Samsung Galaxy SII with Android and TouchWiz UI
(c) Samsung Galaxy Nexus with normal Android operating system
(d) Apple iPhone 4 with iOS from Apple
(e) Nokia Lumina 800 with Windows Phone from Microsoft
(f) Nokia 500 with the Symbian operating system

The operating system element in the XSD is described as follows:

```xml
<xsd:element name="OS">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Android"/>
      <xsd:enumeration value="Android_with_HTC_Sence"/>
      <xsd:enumeration value="Android_with_TouchWiz"/>
      <xsd:enumeration value="iOS"/>
      <xsd:enumeration value="Symbian"/>
      <xsd:enumeration value="WindowsPhone"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

The Schema of the smartphone contains also some elements for the release date of the phone (releaseDate, releaseYear and releaseQuarter). These parameters will only provide a filter for the developed application (see in section 5), to search smartphones that are released after a specific date. In the application you can search for smartphones that have the specific numeric features and/or categorical design features and are released later and are therefore newer. With these parameters you can find out, which manufacturer produces the same smartphones than another manufacturer regarding specific design features. Finally you can see who “steals” the ideas from whom.

Considering the features described before, we now can calculate a similarity value for two smartphones regarding the numerical features, relational features and the categorical features. The other parameters will provide a filter which can be used to search for specific smartphones in the application.

4 Similarity Calculation

The calculation of the similarity of two smartphones depends on the specific features you will compare with. A simple way to calculate the similarity is to use the numerical features.

4.1 Similarity of Numerical Features And Relations

For the similarity of two smartphones, the easiest way is to compare the numerical features `sizeLength`, `sizeWidth`, `sizeHeight` and `displayDimension`, which will describe the metrical design of the phone. The `weight` of the device of course is also a numerical feature, but not so relevant for the visual appearance and will not be considered for now. Let $A$ and $B$ be two smartphones with their numerical features and relations. The similarity of `sizeLength` between $A$ and $B$ will be

$$
similarity_{sizeLength}^{AB} = \begin{cases} 
  \frac{sizeLength_A}{sizeLength_B}, & \text{if } sizeLength_A < sizeLength_B, \\
  \frac{sizeLength_B}{sizeLength_A}, & \text{if } sizeLength_A \geq sizeLength_B. 
\end{cases}
$$
For the numerical relations, the calculation is almost the same. The relation \( \text{sizeWidth} \leftrightarrow \text{sizeHeight} \) is calculated as follows

\[
\text{relation}_{\text{sizeWidth} \leftrightarrow \text{sizeHeight}} = \frac{\text{sizeHeight}}{\text{sizeWidth}}
\]

and the similarity for the relations will be calculated as before

\[
\text{similarity}_{\text{relation}_{\text{sizeWidth} \leftrightarrow \text{sizeHeight}}}^{AB} = \begin{cases} 
\text{relation}_{\text{sizeWidth} \leftrightarrow \text{sizeHeight}}^A, & \text{if } \text{relation}_{\text{sizeWidth} \leftrightarrow \text{sizeHeight}}^A < \text{relation}_{\text{sizeWidth} \leftrightarrow \text{sizeHeight}}^B \\
\text{relation}_{\text{sizeWidth} \leftrightarrow \text{sizeHeight}}^B, & \text{if } \text{relation}_{\text{sizeWidth} \leftrightarrow \text{sizeHeight}}^A \geq \text{relation}_{\text{sizeWidth} \leftrightarrow \text{sizeHeight}}^B
\end{cases}
\]

The similarity of all numerical features and relations will be calculated like

\[
\text{similarity}_{\text{numerical features and relations}}^{AB} = \frac{1}{n} \sum_i \text{similarity}_i^{AB}
\]

where

\( i = \{ \text{sizeLength, sizeWidth, sizeHeight, displayDimension, sizeLength} \leftrightarrow \text{sizeWidth, sizeWidth} \leftrightarrow \text{sizeHeight, sizeLength} \leftrightarrow \text{sizeHeight} \} \) and \( n = 7 \), the count of the numerical parameters used for this similarity calculation. The similarity will now be a value between 0 and 1, where 1 means the phone \( B \) is identical to phone \( A \) and therefore has the perfect similarity (to say \( A \) is equal to \( B \)).

### 4.2 Similarity of Categorical Design Features

The similarity for the categorical design features is pretty easy. The similarity of the feature \( \text{curveCorner} \) for phone \( A \) to phone \( B \) can only reach 3 values.

- \( \text{similarity}_{\text{curveCorner}}^{AB} = 1 \), if the type of \( \text{curveCorner} \) of phone \( A \) is the same type as phone \( B \)’s \( \text{curveCorner} \). Let’s say \( \text{curveCorner}_A = \text{EDGED} \) and \( \text{curveCorner}_B = \text{EDGED} \), then the similarity is equality and therefore 1.

- \( \text{similarity}_{\text{curveCorner}}^{AB} = 0.75 \), if the type of \( \text{curveCorner} \) of phone \( A \) differs only one value as the type of \( \text{curveCorner} \) of phone \( B \). Let’s say \( \text{curveCorner}_A = \text{EDGED} \) and \( \text{curveCorner}_B = \text{EDGED_LOW_ROUNDING} \), the similarity is almost the same, but not equal. You can say, if someone prefers edged corners of the phone, he could also be satisfied, if the corner are almost edged, say a little bit rounded. So for all nearby category values we will get a similarity of 0.75.

- \( \text{similarity}_{\text{curveCorner}}^{AB} = 0 \), else.

The \( \text{similarity}_{\text{curveEdge}}^{AB} \) for design feature \( \text{curveEdge} \) will be calculated as the same as it is done for the parameter \( \text{curveCorner} \).
4.3 Smartphone Similarity

Considering the similarity calculation of the numerical features, the numerical relations and the categorical design parameters, we can now calculate a smartphoneSimilarity$_{AB}$ for all of the features by adding every similarity value like

$$\text{smartphoneSimilarity}_{AB} = \frac{1}{n} \sum_i \text{similarity}_i_{AB},$$

where $A$ and $B$ are the two smartphones that will be compared, $i = \{\text{sizeLength, sizeWidth, sizeHeight, displayDimension, sizeLength} <\text{sizeWidth, sizeWidth} <\text{sizeHeight, sizeLength} <\text{sizeHeight, curveCorner, curveEdge}\}$, and $n = 9$ is the count of the features in $i$ used for the similarity calculation. We now get a value for the smartphones similarity between 0 and 1, indicating how similar the phones are.

4.4 Weighted Similarity Calculation

With the smartphoneSimilarity$_{AB}$ we now have a standard similarity calculation for two smartphones. What if someone’s searching a phone, that has specific parameters like $\text{sizeLength, sizeWidth, sizeHeight}$ and the $\text{displayDimension}$, but he doesn’t care much about the $\text{sizeHeight}$ or some other feature and so this feature is not so relevant. We could adapt the similarity calculation of the numerical features such, that we include a weight for this feature. The weightedSimilarity$_{AB}$ calculation will look like

$$\text{weightedSimilarity}_{AB} = \frac{1}{n} \sum_i (\text{similarity}_i_{AB}) * w_i,$$

where $i = \{\text{sizeLength, sizeWidth, sizeHeight, displayDimension}\}$, $n = 4$ the count of the numeric features and $w_i$ is the weight for the specific numeric feature and is between $[0..1]$, exclusive 0.

With the weightedSimilarity$_{AB}$ we now can modify our calculation (or smartphone search) such, that some features are more relevant than others.

4.5 Filtered Similarity Calculation

The smartphone similarity calculation is finished so far. What we can do now, is to include a filter calculation for the design feature hardkey (see section 3.3) and the filter features from section 3.5. If someone prefers a hardkey or a specific operating system, the similarity calculation (or smartphone search) should also provide such filter mechanism. A filteredSimilarity$_{AB}$ between two smartphones $A$ and $B$ could look like this

$$\text{filteredSimilarity}_{AB} = \text{filter}_{AB} * \text{smartphoneSimilarity}_{AB},$$

where $f$ is the filter and will be calculated as follows
$\text{filter}_{AB} = \{1, \text{ if every filter feature } j \text{ from phone } A \text{ equals the filter feature from phone } B \}

0, \text{ else}

and $j = \{\text{hardkeys}, \text{operatingSystem}, \text{manufacturer}, \}$. The releaseDate of the smartphone could also be a filter, but this one is not an Boolean filter, rather a numeric filter. We could compare the releaseDate of the two phones $A$ and $B$ and could realize if a similar smartphone is newer or older. If you modify the $\text{filter}_{AB}$ such that you will find smartphones from other manufacturers ($\text{manufacturer}_A \neq \text{manufacturer}_B$) than with the releaseDate you could see which manufacturer used the same visual design to produce a smartphone (“Who steals from whom?”).

In the last section I will introduce the developed application for calculating smartphone similarities and search for desired smartphones in the database.

5 Smartphone Similarity Tool

To calculate similarity between smartphones a Smartphone Similarity Tool was developed.

5.1 Requirements and Smartphone Data

The Smartphone Similarity Tool was developed in C# and with Visual Studio 2008. To start the Tool you need the .Net Runtime installed. (Note that the tool was only tested on windows PCs). To prevent problems, the latest .Net Framework 4 (or greater) should be installed. To use the tool, you have to provide some smartphone data in XML format. Appendix A4 shows such an xml file with the data for a HTC Desire HD smartphone. The smartphone xml files should be conform to the specified smartphone.xsd schema file, which is provided by the tool ( and located at /data/smartphone.xsd ) which you can find on the attached CD along with some other smartphone sample data (see appendix A5). The smartphone xml files and smartphone image directory should be copied into the tools data directory. Now you can load the data into the tool ( Menu: File > load data… ). If everything was correct and the xml files are valid and conform to the schema file, the tool should look like in figure 6.

5.2 Searching similar Smartphones

If the smartphone data was loaded, you now can view and switch through the loaded database. If you select a smartphone (Button: Select Smartphone), you can start a search by clicking on the right button “Search Similar Smartphone”.
Now a search will be started by calculating the similarity from every other loaded smartphone to the selected one. This similarity calculation will be done with a predefined standard search (as described in section 4.3). You can modify this search by setting some weight values for the numerical features or activate some filter parameters (see figure 7).

**5.3 Search a new Smartphone**

The similarity search for a smartphone is one way to find similar smartphones. Another way is to search the smartphone database not by calculating similarity regarding a selected phone, but to set some values for the numeric features like $sizeLength = 100$ mm, chose a value for the $curveCorner$ or a specific filter parameter like $manufacturer$. With this kind of search, you can find any similar smartphone that fulfill your desired specifications (Button: Search New Smartphone).

**5.4 Feature Battle – Smartphone Battle**

A third way to search a smartphone is to chose some numerical and categorical features and let them “battle”.

Figure 6: Smartphone Similarity Tool after loading the smartphone xml data.
The tool will give you a choice for all features and you select a value that you would prefer. A preferred value for all selected features will be identified and the database will be searched for smartphones that match these chosen values the best. This is also a way to find smartphones similar to once selected feature values (see figure 8).

On the info tab in the tool, all the features and categories are described to give a quick overview.
Calculating the similarity of smartphones is a good way to see which design is either copied or used from several manufacturers. Not only the numerical features like sizeLength, sizeWidth, sizeHeight and the displayDimension are design parameters, but also the shape of the corners and edges of the phone (curveCorner, curveEdge). With the developed tool you can find a smartphone which looks the same like yours. With all the smartphones on the market and I think there are at least 50 to 60 phones that look almost the same – at first sight – but differ only in small details you wouldn’t recognize first, it would be helpful to have a tool that supports you and eases your decision to find a new one.

Of course the tool is not perfect and could be extended. For example with a web page crawler tool, that would extract the needed smartphone information from the manufacturers web page and automatically create the needed smartphone xml files with all its pictures etc. Also a small tool for creating the smartphone xml file, that you only have to input the data and the tool creates the xml file for you. But for this reason XML was chosen to represent the data because everyone can easily read and

6 Conclusion

Calculating the similarity of smartphones is a good way to see which design is either copied or used from several manufacturers. Not only the numerical features like sizeLength, sizeWidth, sizeHeight and the displayDimension are design parameters, but also the shape of the corners and edges of the phone (curveCorner, curveEdge). With the developed tool you can find a smartphone which looks the same like yours. With all the smartphones on the market and I think there are at least 50 to 60 phones that look almost the same – at first sight – but differ only in small details you wouldn’t recognize first, it would be helpful to have a tool that supports you and eases your decision to find a new one.

Of course the tool is not perfect and could be extended. For example with a web page crawler tool, that would extract the needed smartphone information from the manufacturers web page and automatically create the needed smartphone xml files with all its pictures etc. Also a small tool for creating the smartphone xml file, that you only have to input the data and the tool creates the xml file for you. But for this reason XML was chosen to represent the data because everyone can easily read and
create an xml file for a smartphone, even if they don’t know the whole xml language.
References

Appendix:
A1: Feature: hardkey

hardkey = YES

hardkey = YES

hardkey = NO

hardkey = NO
A2: Feature: curveCorner

curveCorner = EDGED

curveCorner = EDGED
(3D-view)

curveCorner = EDGED_LOW_ROUNDING

curveCorner = EDGED_LOW_ROUNDING
(3D-view)
curveCorner = STRONG_ROUNDING

curveCorner = GLOBAL_ROUNDING

curveCorner = STRONG_ROUNDING

curveCorner = GLOBAL_ROUNDING
A3: Feature: curveEdge

curveEdge = EDGED_NO_ROUNDING_NO_FLATTENING

curveEdge = EDGED_ROUNDING_NO_FLATTENING (3D-view)

curveEdge = EDGED_ROUNDING_LOW_FLATTENING (3D-view)

curveEdge = STRONG_ROUNDING_LONGER_FLATTENING (3D-view)
A4: Smartphone XML Example Data

<?xml version="1.0" encoding="utf-8"?>
<smartphone

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="smartphone.xsd">

<!-- Name -->
<name>HTC Desire HD</name>
<manufacturer>HTC</manufacturer>

<!-- Size -->
<sizeLength>123.0</sizeLength>
<sizeWidth>68.0</sizeWidth>
<sizeHeight>11.8</sizeHeight>
<weight>164</weight>

<!-- Display -->
<displayDimension>4.3</displayDimension>

<!-- Design -->
<hardkeys>false</hardkeys>
<curveCorner>2</curveCorner>
<curveEdge>3</curveEdge>

<!-- Appearance -->
<OS>Android_with_HTC_Sense</OS>
<releaseYear>2010</releaseYear>
<releaseQuarter>4</releaseQuarter>

<!-- Images -->
<imageFront>htc_desire_hd/htc_desire_hd_front.jpg</imageFront>
<imageBack>htc_desire_hd/htc_desire_hd_back.jpg</imageBack>
<imageSide>htc_desire_hd/htc_desire_hd_side.jpg</imageSide>
<imageTop>htc_desire_hd/htc_desire_hd_top.jpg</imageTop>
<imageOverview>htc_desire_hd/htc_desire_hd_overview.jpg</imageOverview>
<imageAllViews>htc_desire_hd/htc_desire_hd_all_views.jpg</imageAllViews>

<price>599</price>
</smartphone>

Data collected from source:
http://www.inside-handy.de/handys/htc_desire-hd_ace/2237_allgemein.html
A5: CD with the Smartphone Similarity Tool
Ich erkläre hiermit gemäß § 17 Abs. 2 APO, dass ich die vorstehende Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

(Datum)       (Unterschirft)