A method for automatic generation of fuzzy membership functions for mobile device's characteristics based on Google Trends

Aitor Almeida, Pablo Orduña, Eduardo Castillejo, Diego López-de-Ipiña
Deusto Institute of Technology - DeustoTech
University of Deusto
Bilbao, Spain
{aitor.almeida, pablo.orduna, eduardo.castillejo, dipina}@deusto.es

Marcos Sacristán
Treelogic
Llanera, Spain
marcos.sacristan@treelogic.com

Abstract— While creating a framework for adaptive mobile interfaces for m-learning applications we found that in order to ease the use of our framework we needed to present the mobile device characteristics to non-expert users in a easy to understand manner. Using fuzzy sets to represent the characteristics of mobile devices, non-expert developers such as teachers or instructional designers can actively participate in the development or adaptation of the educational tools. To be able to automatically generate the fuzzy membership functions of the sets we needed the data of the mobile device market, regrettably this information is not publicly available. To tackle this problem we have developed a method to estimate the market share of each mobile device based on the popularity metrics recovered from Google Trends and then we use that estimated value as the input to generate the fuzzy set of each characteristic. The proposed method allows us to not only model the state of the market in different periods of time, but also to localize the results to adapt them to the mobile market of specific countries. In this paper we will describe the proposed algorithm and we will discuss the obtained results.

Highlights

- Estimating the market share of a mobile device using popularity metrics.
- Automatic generation of membership functions using the device market share and popularity metrics.
- Easing the widespread adoption of new and legacy e-learning applications that use a wide variety of devices.

Keywords- fuzzy; mobile devices; characterization; membership functions; Google Trends; popularity; WURFL;

1. INTRODUCTION

The use of m-learning has increased during the latest years due to the rapid development of the mobile market. Nowadays smartphones are pervasive and mobile internet access is becoming a common occurrence. As students have internet anywhere anytime, instructional designers or teachers have the opportunity to create and use tools to access e-learning material, as well as new approaches for teaching concepts [1] and apply existing e-learning research [2]. New applications have been built using mobile devices features, such as the camera or gyroscopes. Technologies like Augmented Reality for learning [3] become more feasible with the new mobile devices

Existing applications have also been adapted for mobile devices, allowing students to use the systems anywhere. In this area, the authors already adapted existing Remote Laboratories of the University of Deusto to mobile devices [4], so engineering students can access remote experiments that use real hardware[1] from their mobile devices. The task of adapting the design of educational mobile applications to the wide, ever-changing market of mobile devices is complex. Not only the mobile landscape [5] has changed significantly, with iPhone and Android becoming the leading operating systems in less than 4 years since its appearance in the market, but the characteristics of devices supporting a particular operating system change every year. The hardware of the mobile devices is evolving constantly and in such pace that makes difficult for applications to be up to date with recent changes. During the adaptation of the Remote Laboratories of the University of Deusto, it became clear that in order to adapt applications to these constraints, it is required to take into account changes in the market, as well as to be able to track and measure these changes.

To tackle these problems the Imhotep framework [6] was created. The main goal of the framework is to simplify the development cycle of mobile applications as much as possible. This is done by:

- Isolating programmers from hardware constraints of specific device models during the developing phase.
- Simplifying the application building process, allowing to create effortlessly versions for each target device.
- Allowing developers without expertise in mobile development to create more easily mobile applications.

As part of the framework, developers can use preprocessor directives to guide the interface adaptation. These directives employ the device characteristics (screen size, CPU, RAM memory, supported formats, codecs…) to decide how the interface adaptation should be done. One of the identified problems during the development of the framework was that developers without extensive experience usually do not have the required knowledge to identify the exact values to be used in the preprocessor directives. For example a developer might want to know if a CPU is "fast" or if the video capabilities of the device are "good", without having to deal with specific values. Working with values closer to natural language eases the use of preprocessor directives. We encountered initially several problems with this approach. Obviously one mobile device is "fast" when we compare it with the other devices in the market. To have an accurate concept of what is "fast" we have to know the market share of the devices to identify the distribution of the CPU capabilities. Also, the concept "fast" will change over time. What was considered a fast CPU in 2004 nowadays would be considered

[1] http://www.weblab.deusto.es/weblab/

slow. Finally, it will also change with the location; a fast mobile device in a third world country is probably is not so fast in Japan.

To be able to create fuzzy membership functions that model the device characteristics accurately we would have needed the market share data of all the existing devices. Unfortunately this information is not shared by the wireless carriers and we still needed to estimate the position of each device in the market. In this paper we will present a method to infer the device market share from popularity metrics. We will also show how that information can be used to automatically generate membership functions for device characteristics that will be employed in the fuzzy reasoning engine of the Imhotep framework. The interface adaptation process was described previously in [6] and will not be described again in this paper. Our method is based in two different tools. First we have used Wireless Universal Resource FiLe (WURFL) [7], an XML mobile device database, to compile an exhaustive list of the existing devices and their characteristics. With that list we have retrieved the popularity metrics of those devices from Google Trends [8]. Using this data we have developed a process to automatically generate membership functions. We will describe this process in the following paper. In Section 2 we will analyze the related work, in Section 3 we will explain how the generation process works and in Section 4 we will discuss the obtained results. Finally in Section 5 we will expose the conclusions and future work.

2. RELATED WORK

To be able to model the market share of mobile devices and the distribution of each individual characteristic we needed a repository which contained the specifications, capabilities and constraints of those mobile devices. For this purpose we have used the WURFL (Wireless Universal Resource FiLe) database, which is a XML file containing information about the capabilities and technical specifications of thousands of mobile devices. The objective of the WURFL project is to be a central repository of device information for developers to ease the development of mobile applications. The information contained in WURFL has been collected by ScientiaMobile with the collaboration of users and developers around the world. Several authors [9][10][11][12][13] have used previously WURFL to analyze the characteristics of mobile devices.

Google Trends [8] is a tool developed by Google which measures the number of searches that have been done for specific terms in relation to the total number of searches done on Google over time. The results are presented as a graph named Search Volume Index graph. Google Trends has been used to model different domains and scenarios. In [14] authors use data from Google Trends and Google Insights [15] to make short-term economic predictions. This approach has also been used to predict private consumption [16]. Another domain where Google Trends has been used is epidemiological research, studying influenza epidemics [17] or the expansion of Lyme disease [18]. In [19] authors present a web tool for disease outbreak surveillance based on Google Trends. Finally in [20] the trend data is used to track diseases. As can be seen, the information contained in Google Trends is a pool of data that can be used to model, infer and predict the behavior of the users.

Several authors have worked on the problem of adaptive interfaces. Tailoring the presentation of information to the characteristics and constraints of specific mobile phones is important

[21][22]. In [23] and [24] a study of ongoing research in this field is presented, along some examples of adaptive interfaces which use inductive methods to personalize their behavior are described. Besides, the addition of fuzzy logic reasoning allows non expert developers to take part in the development process, which allows working with abstract concepts instead of using specific and probably unknown values of certain features. In other words, fuzzy systems try to describe complex systems with linguistic descriptions, allowing for degrees of membership over the range [0, 1] [25]. Automatic membership function generation has already been addressed by several authors. In [25] and [26] authors describe several methods to automatically generate membership functions using genetic algorithms, which search procedures are based on the mechanics of natural selection. Authors in [27] propose the use of inductive reasoning for the construction of membership functions. In [28] authors use an ad-hoc method to generate the membership functions, while in [29] a neural networks based approach is presented for the same purpose.

3. THE CHARACTERIZATION PROCESS

There are situations where the crisp values of device characteristics are not suitable to be used directly. For example, the developer may want to show certain video only if the screen of the device is “big” or to use a certain reasoning engine only if the processor capabilities are “high”. The main problem with this scenario is that the concept “big” is not directly related to one value and is a relative value (which implies that what is a big screen today probably won’t be big in 2 years). The goal of our system is to identify new capabilities using the already existing ones and to fuzzyfy them. To do this we have defined a set of fuzzy rules that take as input numeric values from the existing capabilities and create symbolic values for the new ones. An example for the reasoning that takes place in this stage would be: “If the resolution is big and the screen size is big the video suitability is very high”. This reasoning will be modelled with fuzzy rules (see Algorithm 1).

```
IF screensize IS big AND resolution IS normal
THEN video IS high;
IF screensize IS big AND resolution IS big
THEN video IS very_high;
```

Algorithm 1. Sample of fuzzy rules

The main problem we have encountered using fuzzy rules is that we need to fuzzify the crisp variables encountered in the databases (in our case WURFL). This raises some challenging questions. What do we consider a “big” screen size? How can we identify what characteristics are inherent of the average mobile device? These concepts are relative to the values of other device models. One screen is big if its height and width are larger than the average values of the other models. To answer these questions we would have to know the actual distribution of the market. Our proposed solution is to use popularity metrics to estimate the market share of the devices (in our case, we use Google Trends). Besides, all the device models can not have the same weight in the calculation, not all the device models have sold the same number of units. This is why the most popular models should have more weight during this calculation. In order to calculate the popularity of one device we have to adjust it with its “age”. Popularity fades with the passing of time. Users tend to change their mobile phones frequently, drastically altering the perception of what is a big screen from one year to another. While this number does not represent the sale volume, it is often used as an indicator of the interest shown by the consumers in a specific model [30]. Due to the lack of data regarding the real sale volume for most mobile devices, it is one of the few available indicators. This trend value can change drastically from one location to another; the most popular devices are not the same in Japan and Europe. To tackle this problem we support the geolocation of the results to filter them according to the needs of the developers.

The device characterization process can be divided into three different steps: the initial data retrieval, the decay process and the automatic membership function generation.

3.1 Initial Data Retrieval

The first step consists on retrieving and formatting all the necessary data. To this end, it is necessary to load the WURFL database containing all the analysed devices and, for each device, retrieve the trends data from Google Trends. This data contains weekly information of how often was a mobile phone name searched in Google since 2004. This information can be searched for a particular country or region or for the whole world.

This data will be used to calculate the importance of each mobile phone. The ideal situation would be having the exact market share of each mobile phone model per country and month, but this information is not public and no accurate public estimation was found for all the 5426 mobile phones models analysed.

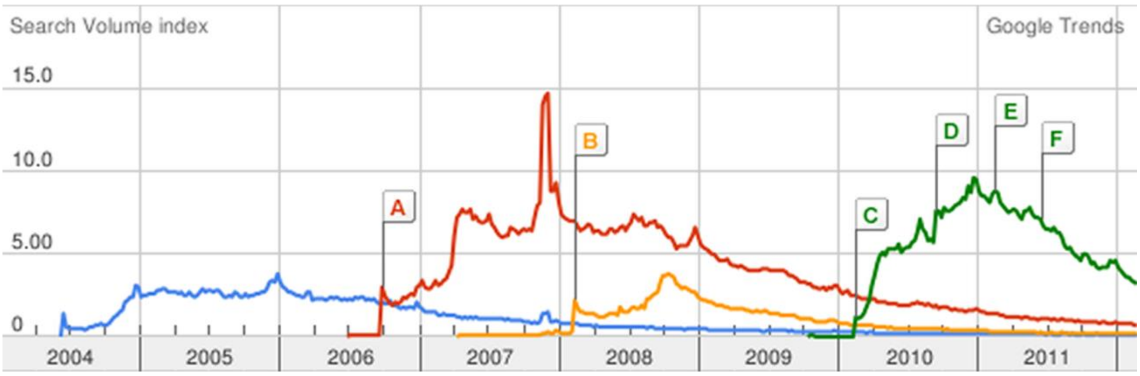


Figure. 1 Trend evolution of four different mobile phones according to Google Trends in the whole world: Nokia 6630 (peak on 2005), Nokia N95 (peak on 2007), Nokia N96 (peak on 2008), HTC Desire (peak on 2010)

On Figure 1 the Google Trends data of different mobile devices is shown and it is possible to check that it matches with the lifecycle of those mobile phones. Due to this match, while Google Trends does not provide an accurate and fully reliable measurement, it can be a valid metric. The drawback of this metric is that it penalizes those mobile phone devices that are never searched for but they are sold (e.g. low end models), and it increases the importance of models which, without an impact on market share, have a special feature (e.g. working under water). However the algorithm presented in this contribution could use other data sources if in the future more accurate data was gathered.

To obtain the data from Google servers, an automated system was created. While Google provides this data for free for research activities, a limit of queries per day is established, so a distributed retrieval system was built. Multiple researchers could help gathering this data in a short range of time (less than a week), and the information of that time is later discarded to avoid penalizations of those mobile phones retrieved first.

3.2 The Decay Process

Google Trends provides information of the trends since 2004. However, nowadays probably a very small percent of mobile phone users use the same mobile phones of that year, so even if a mobile phone was very popular in that year, it should have a smaller weight in the calculation. On the other hand, using only the searches of the very last week is not relevant either, so a period of time must be selected. The problem is that the longer is the selected period of time, the less relevant searches it includes. For instance, if a two years period is selected, the latest months are more relevant than the first ones. Therefore, a decay process that penalizes the oldest results must be established.

The main concern of the decay process is that there is not a perfect one. The ideal situation would be that the market share was public, or at least that estimations were accurate and provided a clear number for all the existing mobile phones. Given that Google Trends is used as a metric based on estimation, there will be no perfect decay process that adapts to each mobile device acting as a corrector of the use of Google Trends.

However, it is possible to mitigate the errors produced by the transitory nature of the market through approximate functions. In imhotep, two strategies have been implemented: the first one, called LogarithmicDecay, uses a logarithmic function to calculate the decay. The value taken for logarithm base is the point where the trends will no longer have any weight in the calculation (5 years in our case) and while the function returns negative values no decay will be applied. Using this decay strategy, five year old trends will not be taken into account and the newer trends will not have any penalty.

The second strategy, called ModelDecay, takes into account the phone plans of the principal telecommunication companies to try to model the mobile phone change cycle among users. In the particular case of Spain, companies selling phones at lower price require 15 months of continuance. The warranty of the mobile phone is established by law in 2 years, and companies start providing better offers to customers who have been with the company during 3 years, although many customers just go to another company to get a new price. While these periods change from country to country and from company to company, they resemble the general trend of mobile devices use. Therefore, the following values are used to calculate the decay in the ModelDecay strategy:

- Latest 15 months: take into account the 100% of the trend value.
- From 16 to 24 months: take into account the 90% of the trend value.
- From 25 to 36 months: take into account the 40% of the trend value.
- From 37 to 60 months: take into account the 10% of the trend value.
- More than 60 months: take into account the 5% of the trend value.

We acknowledge that every user does not change its mobile phone at the end of the acquired mobile plan, and that the assigned percentages are only estimations, and they can be changed in Imhotep if the administrator considers it or finds out more accurate data. We would like to use a more robust model to calculate the decay, but as we will discuss in Section 4 we have found that this is a good approximation. It is therefore a corrector accurate enough to evaluate the rest of the system.

3.3 Automatic Membership Generation

Once we have the processed data (see Figure. 2 for an example of the distribution of the popularity of the different screen resolutions expressed as the multiplication of height and width) and the desired linguistic terms we can automatically generate the membership functions for those terms.

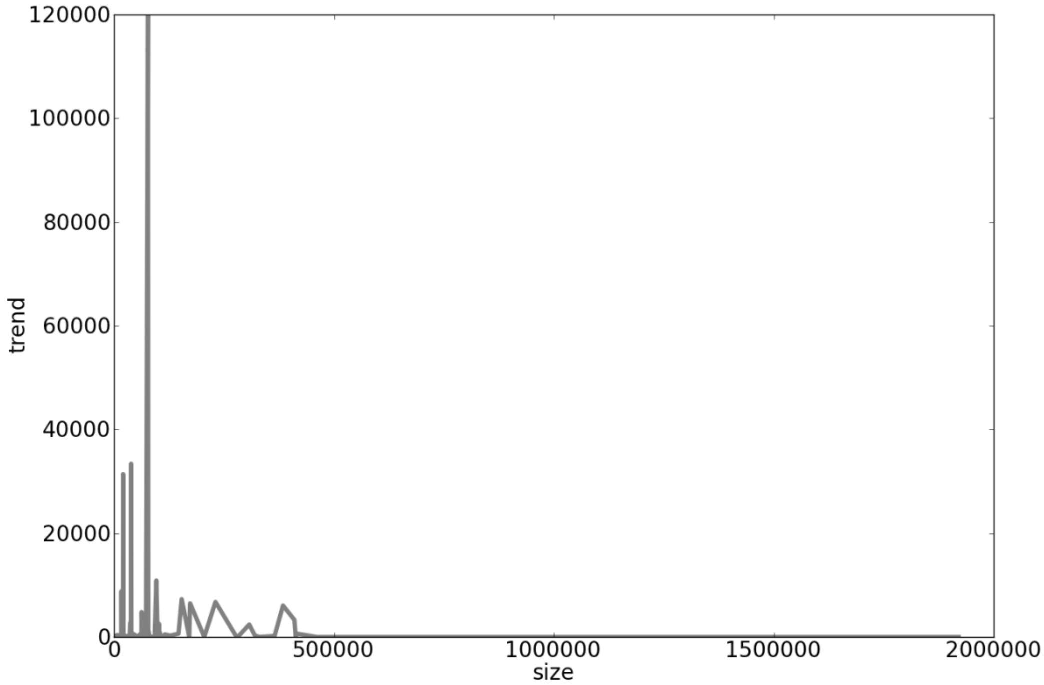


Figure. 2 Worldwide popularity calculation for all the resolutions of the mobile devices in WURFL. Resolutions are expressed as the result of height and wide multiplication, been the highest resolution near the 2000000.

The first step is to divide the data in regions (see Algorithm 2) that will mark the point where each membership function will have its highest value. While creating the regions the algorithm seeks to equally distribute the total trend value contained in each membership function, but usually this goal is not achieved in the first iteration.

```
Function createBaseUniverse
Inputs:
        TRENDS is an association of key-value pairs where the keys are each distinct
        value of the target device's feature and the associated value is the summation of
```

```
        the trends for the devices with that feature value. The keys are ordered
        incrementally.
        NUM_OF_TERMS is the number of desired linguistic terms
Outputs:
        REGION_LIMITS is the location of the regions in the base universe
```

TOTAL_TRENDS ← $\sum_{i=0}^{n} VALUES(TREND_i)$

```
NUM_OF_REGIONS ← NUM_OF_TERMS – 1
IDEAL_TREND ← TOTAL_TRENDS / NUM_OF_REGIONS
ACCUMULATED_TREND ← 0
REGION_LIMITS ← empty set
CURRENT_VALUE ← first feature value in TRENDS

for CURRENT_REGION = 1 to NUM_OF_REGIONS do
        while ACCUMULATED_TREND <= CURRENT_REGION * IDEAL_TREND do
                CURRENT_TREND ← associated trend in TRENDS to CURRENT_VALUE
                ACCUMULATED_TREND += CURRENT_TREND
                CURRENT_VALUE ← next feature in TRENDS
        end while
        add CURRENT_VALUE to REGION_LIMITS
end for
return REGION_LIMITS
```

Algorithm 2. Calculation of the base universe

To solve this problem we generate every possible permutation by moving each of the initial region boundary one step to each side. For each universe we calculate its deviation from the ideal one (see Algorithm 3). First we discard inconsistent universes following these rules:

- If the first region in the universe does not start in the 0 point, then that universe is discarded.
- If the left boundary of a region in the universe starts after the right boundary, then that universe is discarded.
- If the left boundary of a region starts before the right boundary of a previous region, then that universe is discarded.

The second step is to calculate the deviation of each remaining universe. What we seek is to minimize the deviation from the ideal universe, thus, we select the universe with the lowest deviation.

```
Function findBestUniverse
Inputs:
        REGION_LIMITS is the location of the regions in the base universe
Outputs:
        BEST_UNIVERSE is the universe with the best fitness

ALL_UNIVERSES ← every possible permutation generated by moving one step the limits
contained in REGION_LIMITS

for each PERMUTATION in ALL_UNIVERSES do
        if first region of PERMUTATION does not start in 0 then
                remove PERMUTATION from ALL_UNViVERSES
        for each REGION in PERMUTATION do
                if left limit of REGION > right limit of REGION then
                        remove PERMUTATION from ALL_UNIVERSES
                elif left limit of REGION < right limit of previous REGION then
```

```
                        remove PERMUTATION from ALL_UNVIERSES
        end for each
end for each

UNIVERSE_FITNESS ← empty set

for each PERMUTATION in ALL_UNIVERSES
        PERMUTATION_FITNESS ← ∑_{region∈PERMUTATION}|IDEAL_TREND − ∑_{trend ∈region} trend|
        add PERMUTATION_FITNESS, PERMUTATION to UNIVERSE_FITNESSES
end for each
BEST_UNIVERSE ← first PERMUTATION in UNIVERSE_FITNESS with lowest PERMUTATION_FITNESS

return BEST_UNIVERSE
```

Algorithm 3. Calculation of the best possible universe

Once we have found the best universe we can finally build the membership functions for each linguistic term (see Algorithm 4). To do this we take into account that:

- The region boundaries mark the inflexion point from the ascending and descending curves of a linguistic term.
- The first linguistic term will only have a descending curve that will start in the left boundary of the first region and will end in the right boundary of the first region.
- The last linguistic term will only have an ascending curve that will start in the first boundary of the last region and will end in the right boundary of the last region.
- The ascending curves are calculated accumulating the trend values in a region.
- The descending curves are symmetrical to the ascending curves: dc(x) = 1 – ac(x), where ac is the ascending curve and dc the descending curve.

```
Function calculateMembershipFunctions
Inputs:
BEST_UNIVERSE is the universe with the best fitness
Outputs:
        MEMBERSHIP_FUNCTIONS are the membership functions for the linguistic terms

LAST_CURVE ← empty set
MEMBERSHIP_FUCTIONS ← empty set
for each REGION in BEST_UNIVERSE do
        ASCENDING_CURVE ← ac(x) = ∑_{i=left_limit}^{x} TREND_i where x ∈ feature values in REGION
        DESCENDING_CURVE ← dc(x) = 1 − ac(x) where x ∈ feature values in REGION
        CURRENT_CURVE ← LAST_CURVE + DESCENDING_CURVE
        add CURRENT_CURVE to MEMBERSHIP_FUNCTIONS
        LAST_CURVE ← ASCENDING_CURVE
end for

add LAST_CURVE to MEMBERSHIP_FUNCTIONS
return MEMBERSHIP_FUNCTIONS
```

Algorithm 4. Calculation of the membership functions

The result of this process can be seen in Figure. 3. Using the data shown in Figure. 2 we have divided it in 3 linguistic terms: small resolutions (dashed line), normal resolutions (solid line) and big resolutions (dotted line).

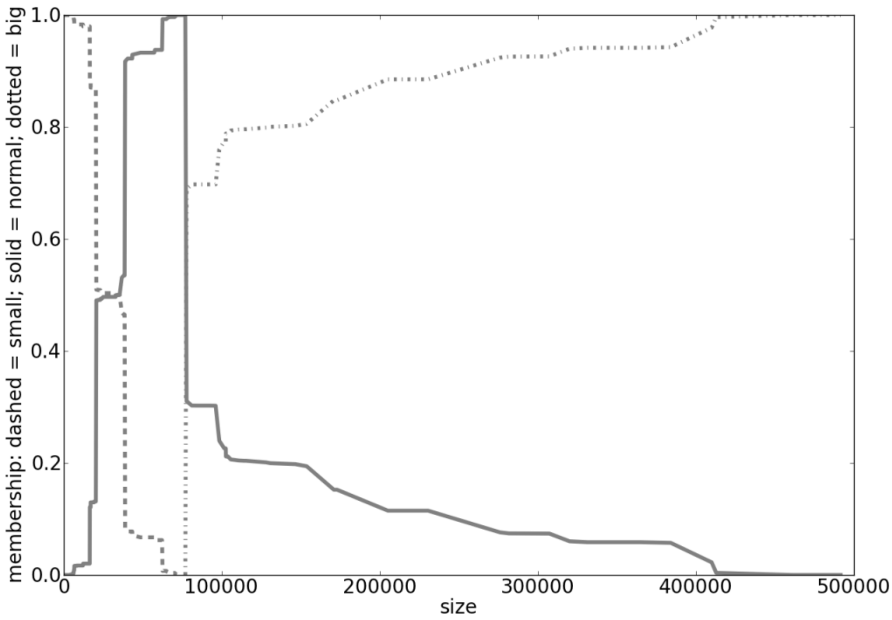


Figure. 3 Generated membership functions for the screen resolution

4. OBTAINED RESULTS

Due to the lack of hard data regarding the market share of each device it is difficult to evaluate the results, but we can assess the *goodness of fit* of the created model comparing different results. In the first test we compare the resolution size of different devices with worldwide popularity data taken until April 2010 (see Table 1). We have divided the screen resolution into 3 linguistic terms: small, normal and big.

As can be seen on the most common resolution worldwide on March 2010 according to our system was 240x320 pixels. We have repeated the tests, but this time we only used the Nokia N95 and N97 as target devices and we have changed the location of the trends. The generated membership functions change from one location to another. While the results for the worldwide and Spanish markets are similar the membership functions generated for the Japanese market show some changes. The main difference is that there is a significant number of resolutions that can be considered mainly "big" worldwide and in Spain that are considered mainly "normal" in Japan.

Device	Resolution	Membership value for each term
Nokia 7110	96 x 65 pixels (4224)	small: 0.91073068 normal: 0.08926932 big: 0
Nokia 6630	176x208 pixels (36608)	small: 0.72814896 normal: 0.27185104 big: 0
Nokia N95	240x320 pixels (76800)	small: 0 normal: 1 big: 0
HTC Hero	320 × 480 pixels	small: 0 normal: 0.19230388

	(153600)	big: 0.80769612
Apple IPhone 3G	320 × 480 pixels (153600)	small: 0 normal: 0.19230388 big: 0.80769612
Nokia N97	640×360 pixels (230400)	small: 0 normal: 0.09409409 big: 0.90590591
Apple IPad	1024 × 768 pixels (786432)	small: 0 normal: 0 big: 1

Table 1 Membership values for different devices using worldwide data

This is more easily seen in Table 2. Comparing the results of the membership values for the Nokia N95, HTC Desire (Android) and Blackberry 8703e in the three different markets we can see that resolutions in Japan tend to be bigger than worldwide and Spanish mobile device's resolutions.

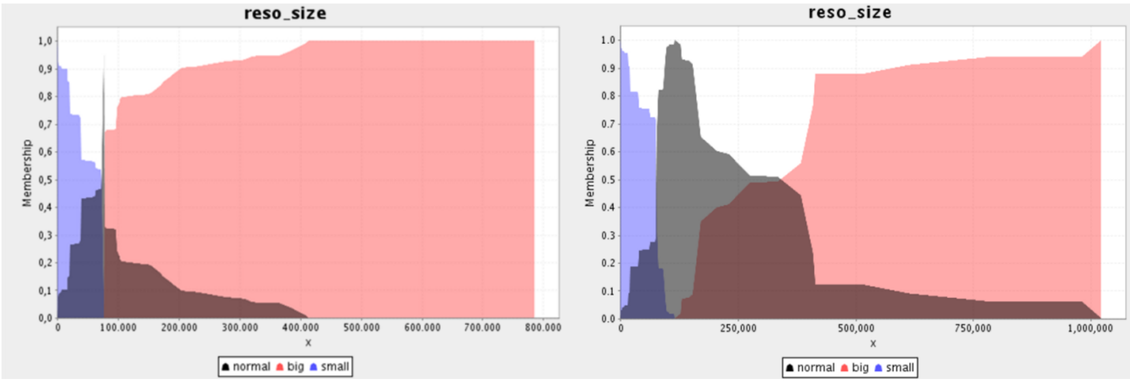


Figure. 4 Changes in the resolution size membership functions in two different periods. Intersections mark those points where a resolution is member of two linguistic terms with the same value

The differences in trend values in different instants can be seen in Figure. 4. We can see how during a one year period (from February 2010 to April 2011) resolutions that where considered mostly "big" changed and become "Normal". In Figure. 5 this evolution can be seen more easily. The figure shows how the membership values for each linguistic term (small, normal and big) of the resolution size change from early 2005 to early 2011 for two different mobile phones. The vertical line indicates the release date of the mobile phone. Using the collected trend data and our system we can infer how "big" or "small" the resolutions of the devices were on a specific date, even before the commercialization of the specific mobile phone model. Graphics show how initially the resolution of those devices was considered normal or even big, but with the appearance of new and better mobile phone it ended been considered small.

Device	Location	Membership value for each term
Nokia N95	Worldwide	small: 0 normal: 1 big: 0
Nokia N95	Spain	small: 0 normal: 1

		big: 0
Nokia N95	Japan	small: 0.66996612 normal: 0.33003388 big: 0
HTC Desire	Worldwide	small: 0 normal: 0.13444257 big: 0.86555743
HTC Desire	Spain	small: 0 normal: 0.4423023 big: 0.5576977
HTC Desire	Japan	small: 0 normal: 0.72208264 big: 0.27791736
Blackberry 8703e	Worldwide	small: 0 normal: 1 big: 0
Blackberry 8703e	Spain	small: 0.36131458 normal: 0.63868542 big: 0
Blackberry 8703e	Japan	small: 0.94690347 normal: 0.05309653 big: 0

Table 2 Membership values for different markets

5. CONCLUSIONS

While developing the Imhotep Framework we created a set of preprocesor directives for adaptable interfaces. These directives allow to adapt the interface of mobile applications taking into account the characteristics of the target device. To be more easily used by non-expert users we wanted to express these characteristics with values as close as possible to natural language. One of the problems with this approach was the lack of public data about the market share of each mobile device. In this paper we have described how to estimate the market share of a mobile device using popularity metrics and how to automatically generate fuzzy membership functions using that data.

We have also shown the results of this process, comparing the results of different mobile phones and different locations, and showing how the passing of time changes the relative perception of the characteristics of the devices. In Section 4 we have explained how the same mobile phone gets different membership values depending on the region, showing how the state of the mobile market influences the perceptions about the characteristics of mobile devices. As future work we would like to implement some new decay functions and compare the results with the existing ones. Our aim is to create a more sound and robust decay model for the mobile device market. We will also implement a new version of the membership function generation mechanism that will allow users to specify the percentage of the trends contained in each linguistic term.

ACKNOWLEDGMENT

This work has been supported by project grant TSI-020301-2008-2 (PIRAmIDE), funded by the Spanish Ministerio de Industria, Turismo y Comercio. We would like to thank the members of our laboratory that helped us recovering the trend data.

REFERENCES

[1] Lee Chao (Editor). Open Source Mobile Learning. 2011. ISBN: 978-1-60960-613-8.

[2] Sampson, D.G., Lytras, M.D., Wagner, G., Díaz, P. Ontologies and the Semantic Web for E-Learning. Educational Technology & Society, vol. 7, n. 4, pp 26-28. 2004.

[3] Liu, T.Y., Tan, T.H., Chu, Y.L. QR Code and Augmented Reality-Supported Mobile English Learning System. Mobile Multimedia Processing, pp 33-52. 2010.

[4] Javier Garcia-Zubia, Diego López-de-Ipiña, Pablo Orduña. Mobile Devices and Remote Labs in Engineering Education 8th IEEE International Conference on Advanced Learning Technologies, ICALT2008. Santander (España). Jul. 2008.

[5] Butler, M. Android: Changing the mobile landscape. IEEE Pervasive Computing, vol 10, n. 1, pp 4-7. 2011.

[6] Aitor Almeida, Pablo Orduña, Eduardo Castillejo, Diego Lopez-de-Ipiña, Marcos Sacristan. Imhotep: an approach to user and device conscious mobile applications Personal and Ubiquitous Computing (Journal), Volume 15, Number 4, pages 419-429. Springer. ISSN: 1617-4909. DOI: 10.1007/s00779-010-0359-8. April 2011.

[7] WURFL, the Wireless Universal Resource File. http://wurfl.sourceforge.net/

[8] Google Trends. http://www.google.com/trends

[9] S. Gómez, C. Mejía, D. Huerva, y R. Fabregat. Context-aware adaptation process to build units of learning based on IMS-LD standard. Proceedings of the International Conference on Education and New Learning Technologies, 2009.

[10] Georgieva, E. and Georgiev, T. Methodology for mobile devices characteristics recognition. Proceedings of the 2007 international conference on Computer systems and technologies, 2007.

[11] Zhao, X. and Anma, F. and Ninomiya, T. and Okamoto, T. Personalized Adaptive Content System for Context-Aware Mobile Learning. Int. J. of Computer Science and Network Security, volume 8, number 8, pp 153-161, 2008.

[12] Eisinger, R. and Manzato, M.G. and Goularte, R. Devices descriptions for context-based content adaptation. Web Congress, 2005. LA-WEB 2005. Third Latin American, 2005.

[13] Glover, T. and Davies, J. Integrating device independence and user profiles on the Web. BT Technology Journal, volume 23, number 3, pp 239-248, 2005.

[14] Varian, Hal R. and Choi, Hyunyoung, Predicting the Present with Google Trends (April 2, 2009). Google Research Blog http://googleresearch.blogspot.com/2009/04/predicting-present-with-google-trends.html. Available at SSRN: http://ssrn.com/abstract=1659302

[15] Google Insights. http://www.google.com/insights/search/#

[16] Schmidt, T. and Vosen, S. Forecasting Private Consumption: Survey Based Indicators Vs. Google Trends. RUB, Dep. of Economics. 2009.

[17] Ginsberg, J., Detecting influenza epidemics using search engine query data. Nature, vol. 457, number 7232, pages 1012—1014, 2008.

[18] Seifter, A. and Schwarzwalder, A. and Geis, K. and Aucott, J. The utility of "Google Trends" for epidemiological research: Lyme disease as an example. Geospatial Health, vol 4, num 2, pp 135-137. 2010.

[19] Carneiro, H.A. and Mylonakis, E. Google trends: a web-based tool for real-time surveillance of disease outbreaks. Clinical infectious diseasesvol 49, num 10, pp 1557. 2009

[20] Valdivia, A. and Monge-Corella, S. Diseases Tracked by Using Google Trends, Spain. Emerging Infectious Disease vol 16, num 1, pp 168. 2010.

[21] G. Buchanan, S. Farrant, M. Jones, H. Thimbleby, G. Marsden, y M. Pazzani, «Improving mobile internet usability», in Proceedings of the 10th international conference on World Wide Web, 2001, pp. 673–680.

[22] D. Billsus, C. A. Brunk, C. Evans, B. Gladish, y M. Pazzani, «Adaptive interfaces for ubiquitous web access», Communications of the ACM, vol. 45, nº. 5, pp. 34–38, 2002.

[23] P. Langley, «Machine learning for adaptive user interfaces», in KI-97: Advances in artificial intelligence, 1997, pp. 53–62.

[24] P. Langley y H. Hirsh, «User modeling in adaptive interfaces», COURSES AND LECTURES-INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES, pp. 357–370, 1999.

[25] Homaifar, A. and McCormick, E. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. Fuzzy Systems, IEEE Transactions on, vol 3, num 2, pp 129-139. 1995.

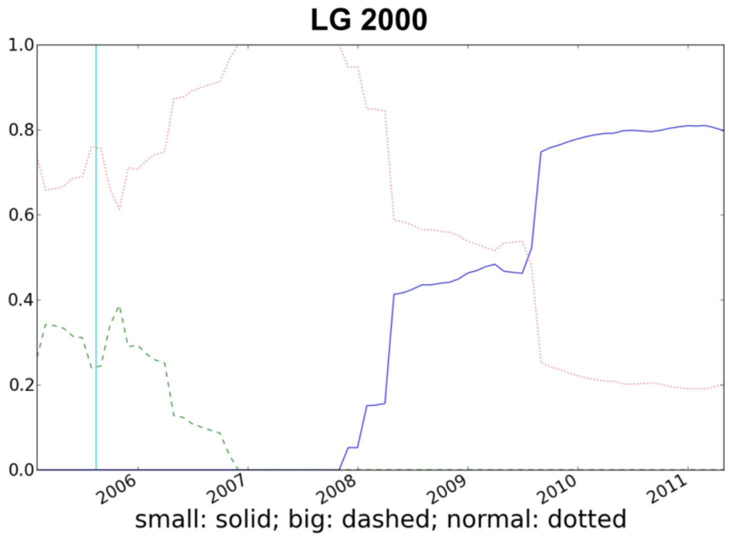
[26] Shimojima, K. and Fukuda, T. and Hasegawa, Y. Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm. Fuzzy sets and systems, vol 71, num 3, pp 295-309. 1995.

[27] Kim, CJ and Russell, BD. Automatic generation of membership function and fuzzy rule using inductive reasoning. Industrial Fuzzy Control and Intelligent Systems, 1993., IFIS'93., Third International Conference on, pp 93-96. 1993

[28] Nieradka, G. and Butkiewicz, B. A method for automatic membership function estimation based on fuzzy measures. Foundations of Fuzzy Logic and Soft Computing, pp 451—460. 2007.

[29] S. Wu, M. J. Er, y Y. Gao, «A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks», Fuzzy Systems, IEEE Transactions on, vol. 9, nº. 4, pp. 578–594, 2001.

[30] Xu, Kaiquan; et al, "Predict Market Share with Users' Online Activities Data: An Initial Study on Market Share and Search Index of Mobile Phone" (2010). PACIS 2010 Proceedings.



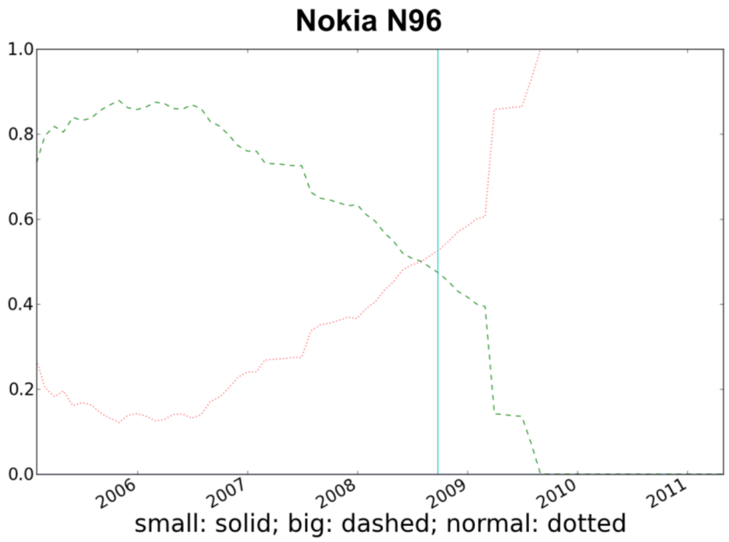


Figure. 5 Evolution of the membership functions of the resolution size of LG C200 and Nokia N96 mobile phones from early 2005 to early 2011. The vertical line shows when the mobile phone was launched