Journal of Software Engineering and Applications, **2013, 6, 582-588**
Published Online November 2013 (http://www.scirp.org/journal/jsea)
http://dx.doi.org/10.4236/jsea.2013.611070

An Improved Storage Structure Format for VCF Standard

Qi Cao, H. J. Cai[*], Tianqi Cai

International School of Software, Wuhan University, Wuhan, China.
Email: [*]hjcai@whu.edu.cn

Received October 6, 2013; revised October 30th, 2013; accepted November 6th, 2013

ABSTRACT

The Visiting Card Format (VCF) has already been widely used in mobile devices and personal computers (PC). Many mobile manufacturers have developed visiting card recognition systems by using Optical Character Recognition (OCR) technology, which can convert and save the information on the paper cards. However, the inaccuracy of OCR always leads to error in recognition, and it only seizes text information, ignoring images. Our research is to solve that problem based on the vCard format by creating a new VCF, vCardPackage, which contains both textual and pictorial information. It consists of a layout layer, a text content layer, a resource layer and a template layer. The design combines content, layout and images together, and enables users to view or modify mobile devices. The XML storage structure is adopted in the vCardPackage, which is compatible with VCF data, so that the original data can be stored completely when shared or transferred.

Keywords: Component; VCF; Card; OCR

1. Introduction

This article is based on vCard format. A vCard presents a standard visiting card format which is widely used in PC and mobile devices. It is a string-based type of label formats and covers most of the visiting card information.

Our research starts from that format and designs a new visiting card format that combines visiting card content, layout and pictures, completes presentation and editing systems on mobile devices, according to the characteristics and actual demand of the mobile Internet communication. By that, the original data can be completely saved in transmission.

The next chapter describes the background of this paper, mainly on former works and related theories on this kind of visiting card communication format.

2. Related Work

Versitcard was originally proposed in 1995 by the Versit Consortium, which consisted of Apple, AT & T Technologies (later Lucent), IBM and Siemens. It is a trend to use the electronic equipment as the carrier to take advantage of the VCF format information exchange visiting cards. How to make this exchange of information is more convenient and optimized, is worth exploring. Lots of work has been done.

2.1. OCR

Optical character recognition, usually abbreviated to OCR [1], can scan and convert handwritten, typewritten or printed text into machine-encoded text (VCF Format) and store the data locally ,thus leaving out the step of manual input and speeding the conversion [2].

2.2. RFID

Radio-Frequency Identification (RFID [3]) is a wireless non-contact system that uses radio-frequency electromagnetic fields to transfer data from an attached tag to an object, for the purposes of automatic identification and tracking. Some tags require no battery and are powered by the electromagnetic fields used to read them. So it is convenient and accelerated to transmission during the unlimited devices [4].

However, that mode only deals with the transmission speed and convenience. According to the analysis of the implementation process and its characteristic, we have drawn the conclusion that there are some key points to deal with in this paper: the recognition result after the preparation.

[*]Corresponding author.

Shortcomings of the above conversion are fatal. We decide not to convert, but store a custom electronic visiting card directly which can achieve the effect of the paper visiting card. It requires a definition of a visiting card format to display pictures, rich text and small enough. XML comes to our mind firstly.

XML is a source language used to tag electronic documents and make them structural, which allows a user to define the markup language. The development of an XML parser is comparatively simple. For simplicity, exquisite and satisfaction, we have chosen XML as the source language of the format of our visiting cards.

3. Format Definition

The idea of the whole format definition is defining a new card format, Visiting Card Package (VCP), to make data compatible under VCF format, VCP file is based on XML standard [5]. The root tag of a VCP format is that VCP represents a set of cards, with at least one vSlide, also contains vMaster and vBackground. The vSlide includes the card's unique image or text, such as a specific address, telephone number, email, text information are stored in the external VCF file and text styles are stored in the vSlide external RTF [6-17] file. The vMaster acts as a model, contains images and texts, and usually stores some common pictures or texts compared to vSlide, such as background, company logo, easy to reuse. The vBackground only stores images used for background. **Figure 1** shows a card demo with vSlide, vMaster and vBackground. The following coordinate-related illustration in format definition is original at vSlide's upper-left corner (the point O in **Figure 1**), the domain contains two domains. One is the text domain, which is a rectangular area in continuous text of VCP. The other is the picture domain, a rectangular area in a complete picture of VCP.

Whole structure of VCP is depicted in **Figure 2**.

Corresponding XML schema is described in **Figure 3**.

We have defined some basic tag type for vSlide vMaster and vBackground: vLocation, vSize, vCard, vText and vImage.

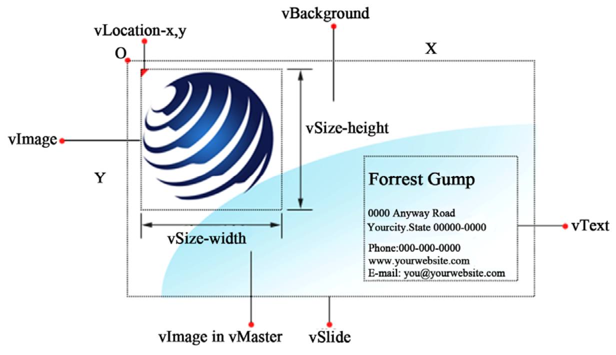

Figure 1. A complete card demo.

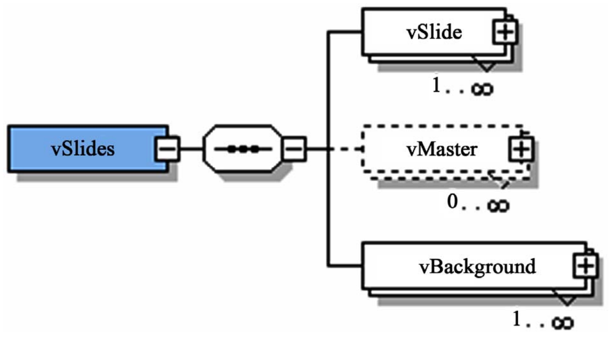

Figure 2. Whole structure of VCP.

```
<xs:element name="vSlides">
      <xs:complexType>
            <xs:sequence>
                  <xs:element name="vSlide"
                        type="vSlide" id="slide" mi-
                        nOccurs="1" maxOc-
                        curs="unbounded" />
                  <xs:element name="vMaster"
                        type="vMaster"   id="master"
                        maxOccurs="unbounded"/>
                  <xs:element name="vBackground"
                        type="vBackground"
                        id="background" maxOc-
                        curs="unbounded"/>
      </xs:sequence>
        </xs:complexType>
</xs:element>
```

Figure 3. XML schema of VCP.

3.1. vLocation

For the control of its domain location (text domain, picture domain) in the mobile phone screen view, its frame is displayed in **Figure 4**.

The XML schema definition is shown in **Figure 5**.

In **Figure 5**, both x and y are integers, representing deviation on x and y coordinate respectively. While z is another integer, if the area of the two domains conflict, the one with bigger z value cover the small one.

3.2. vSize

For the control of the wide, height of its domain, the frame is given in **Figure 6**.

XML schema definition is shown as **Figure 7**.

In **Figure 7**, width and height are 2 integers, representing the width and height of the domain respectively.

3.3. vCard

Used to locate the plain text file path stored in vSlide and vMaster, its frame is depicted in **Figure 8**.

XML schema definition is given in **Figure 9**, in which *file* is a string, representing referred VCF file path. The specified file stores all the plain text without style in

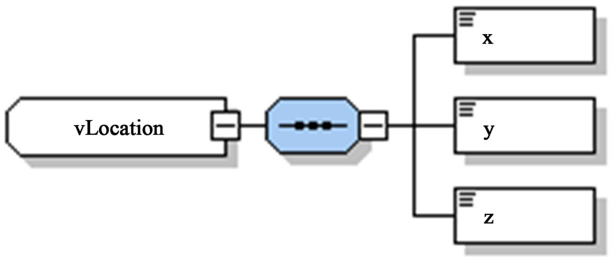

Figure 4. Constructor of vLocation.

```
<xs: complex Type name="vLocation">
      <xs:sequence>
            <xs: element name="x"
                  type="xs:integer"/>
            <xs: element name="y"
                  type="xs:integer"/>
            <xs: element name="z"
                  type="xs:integer"/>
      </xs: sequence>
</xs: complex Type>
```

Figure 5. XML schema of vLocation.

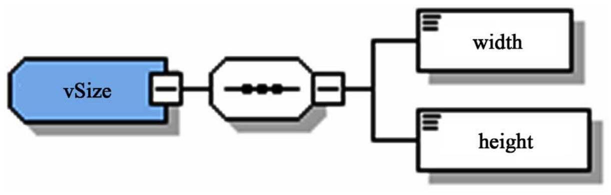

Figure 6. Constructor of vSize.

```
<xs: complexType name="vSize">
      <xs:sequence>
           <xs: element name="width" type="xs: in-
                 teger"/>
           <xs: element name="height" type="xs:
                 integer"/>
      </xs: sequence>
</xs: complexType>
```

Figure 7. XML schema of vSize.

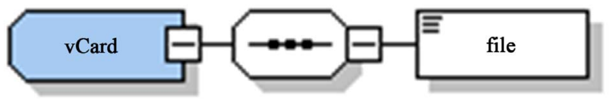

Figure 8. Constructor of vCard.

vCardPackage.

3.4. vImage

To manipulate the picture illustration in vSlide, vMaster, and vBackground, a frame is displayed in **Figure 10**.

The XML schema definition is given in **Figure 11**.

```
<xs:complexType name="vCard">
      <xs:sequence>
            <xs:element name="file"
                  type="xs:string"/>
      </xs:sequence>
</xs:complexType>
```

Figure 9. XML schema of vCard.

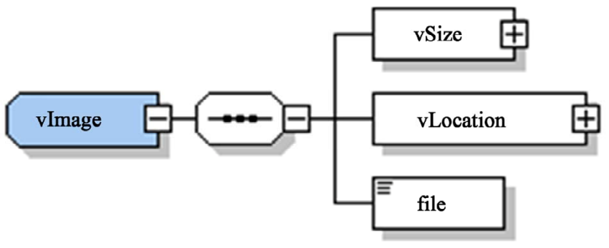

Figure 10. Constructor of vImage.

```
<xs:complexType name="vImage">
      <xs:sequence>
            <xs:element name="vSize"
                  type="vSize"/>
            <xs:element name="vLocation"
                  type="vLocation"/>
            <xs:element name="file"
                  type="xs:string"/>
      </xs:sequence>
</xs:complexType>
```

Figure 11. XML schema of vImage.

The *file* is a string, setting up the file path of the picture to be filled in with. For instance, the left corner picture in **Figure 1**, the image path is the image_0.png file under the images folder in the current directory, and set its initial coordinates as: x = 10, y = 10, z is default, width is 150, height is 150, then the codes should be as shown in the **Figure 12**.

3.5. vText

For the control of text illustration in vSlide and vMaster, the frame is shown in **Figure 13**.

XML schema definition is given in **Figure 14**.

The vText's role is to control the text area in vSlide, vMaster, and a card must have several vText to show the key information. vSize controls t vText's size, vLocation controls its coordinates in vSlide, vMaster, and then display the text in the rich-text file specified by document .The rich-text file is a improved RTF file, which stores the text index (the text in VCF file specified by vCard) and text corresponding style. Take the lower right corner vText in **Figure 1** as an example, we want to set its surname, given name as bold, then the vText label

```
<vImage>
    <vSize>
        <width>150</width>
        <height>150</height>
    </vSize>
    <vLocation>
        <x>10</x>
        <y>10</y>
        <z>0</z>
    </vLocation>
    <file >image/image1.png</file>
</vImage>
```

Figure12. Code of vImage sample.

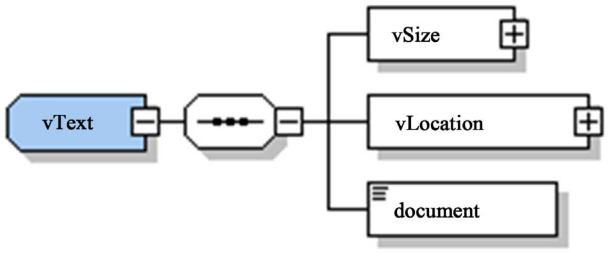

Figure 13. Constructor of vText.

```
<xs:complexType name="vText">
    <xs:sequence>
        <xs:element name="vSize"
            type="vSize"/>
        <xs:element name="vLocation"
            type="vLocation"/>
       <xs:element name="document"
           type="xs:string"/>
    </xs:sequence>
</xs:complexType>
```

Figure 14. XML schema of vText.

```
<vText >
    <vSize>
        <width>170</width>
        <height>70</height>
    </vSize>
    <vLocation>
        <x>220</x>
        <y>120</y>
        <z>0</z>
    </vLocation>
    <document>main.rtf</document>
</vText>
```

Figure 15. Code of vText sample.

```
<vText >
    <vSize>
        <width>170</width>
        <height>70</height>
    </vSize>
    <vLocation>
        <x>220</x>
        <y>120</y>
        <z>0</z>
    </vLocation>
    <document>main.rtf</document>
</vText>
```

Figure 16. Code of part of improved rtf file.

code and part of RTF code should be as in **Figures 15** and **16**.

VCP's three main labels are formed by the basic label defined above.

4. Parsing and Rending Algorithm

In **Figure 17** a user processing the VCP is described.

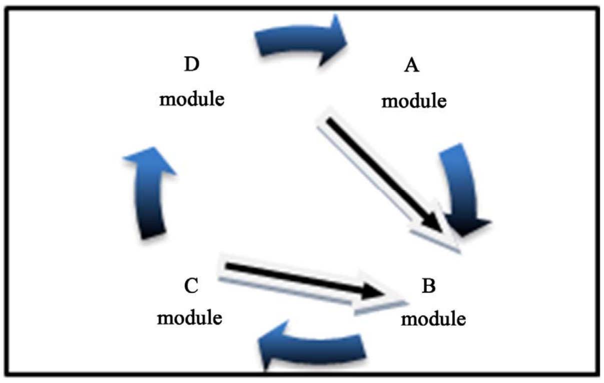

Figure 17. System process modules.

4.1. OCR Pattern Recognition Module

The OCR identification of card using Tesseract OCR Engine, for the identification of text will be generated VCF format automatically and added to the address book which supports VCF format, if there are some errors, we need to manually modified through OCR technology can determine each text block coordinate, and then if the text has been added to VCF file, the text will remove form the origin. The rest should add to vMaster inside.

The overall process as the follow:

1) Take photographs of the file;

2) Identify generation engine;

3) Generate VCP file.

Related algorithm in the resolution process used to as following.

- Text Recognition and Conversion Algorithm

1) Parsing a line of text, the counter is incremented by 1. If any VCF keywords are included, then create a document in the text label and associated with an RTF document storage and physical location information.

2) The identified VCF information will be records to VCF file and the above RTF will be set reference by VCF content. Recording the relative position of the lines of text information, establish vLocation tag, length and width information, establish vSize tags.

3) By vLocation, vSize, document tag data add vText tag, then to add to the vText tag vSlide tag. It defined as read by vLocation, vSize corresponding position of the rectangle.

- Picture Identification Conversion Algorithm

1) The user selected image area, the counter is incremented by 1.

2) Based on the user-selected area of the screen to generate vLocation vSize tags, the selected area will be saved as a picture, and save the physical path to the file tags.

3) Add the vLocation, vSize, file information the vImage tag, then to vImage tag added to vSlide tag. It defined as read by vLocation, vSize corresponding position of the rectangle.

The Position Algorithm.

If need to adjust the position and size of the vText or vImage, drag the object, with the new Location LocationE, if (vTextM.vLocationE, vTextM.vLocationE + vSizeD) z-index conflict with the other regions, the vTextM the z-index where the maximum on the z-index increment, and then update the canvas.

- Zoom Out and In Algorithm

If we zoom in to corresponding vText, the vSize of vTextM will update with new value. If (vTextM. vLocationC, z-index conflicts with other regional vTextM. vLocationC + vSizeH) is vTextM z-index where the maximum z-index increment, and then with the new canvas.

According to the above four algorithms, specific algorithm can be summarized as follows.

1) Mobile phone camera shot and take pictures of visiting cards;

2) Identification pictures via OCR engine;

3) Create a whole file VCP file.

Create a new empty VCP file vCardPackage.xml the root label of vCardPackage vSlides.

a) Create the visiting card vSlide an empty label to add to vSlides child nodes identify the size of a visiting card out of add vCardSize label, means the size of a visiting card;

b) Create an empty the VCF file vcf1.vcf ();

c) Add the path of VCF to vSlide vCard tag, then add vCard tag to vSlide inside;

4) Establishing the visiting cards of data files vSlide;

VCF text information on the identification card to achieve the following:

a) Transformed according to the text recognition algorithm, the calculation;

b) If a text is not empty continue first step in the operation, otherwise exit.

VCF text recognition, begin to identify the picture on the visiting card information, the specific process is as follows:

a) According to the picture transformation algorithm to calculate;

b) If the user chooses to continue to select the image, continue to the first step, otherwise quit parsing.

Identify the complete text and pictures, records related readable area, complete generation of vSlide tag data;

5) File vMaster creation

Data locked outside the region of the identification vSlide process identification to create a new empty the VCF file vcf1.vcf ();

Template text recognition module, the corresponding process is as follows:

a) Transformed according to the text recognition algorithm, the calculation;

b) If a text is not empty continue first step in the operation, otherwise exit.

The template image recognition module.

a) According to the picture transformation algorithm to calculate;

b) If the user chooses to continue to select the image, continue to the first step, otherwise quit parsing.

Uniform preservation does not recognize the text and pictures as the background image:

a) In accordance with screen size area to generate vLocation and the vSize, will not read the whole area is saved as pictures, save the physical path to the file tags. The selected area on behalf of the entire visiting cards;

b) vLocation, vSize, file information add vImage label, then to vImage label added to vMaster label;

c) According to vLocation, vSize corresponding position of the rectangle defined as read.

6) To detect whether the entire card data has been completed to read, if to read the finish will vSlide and vMaster labels add to vSlides label.

7) If it continues to recognize the other card, continue to identify and production vSlide tab and vMaster of labels, and then add to vSlide label.

4.2. VCP Rendering Module

1) Use zipInputStream VCP file is read into memory;

2) First read vSlides notes, get which vSlide number of, traverse vSlide label;

3) For a single vSlide to be associated vMaster the foreign key vMarsterId, and associated the vBackground foreign key vBackgroundId ;

4) Out-key vBackgroundId vBackground, parse inside vImage the the the vImage inside of vSize, vLocation file to draw the corresponding picture, as the lowest level of the draw;

5) Foreign key vMasterId, get vMarster, parsing the inside of vText and vImage to draw which vText vSize,

vLocation and document the rtf the text, and then according to the the vImage inside of vSize vLocation, file to draw the corresponding picture as an intermediate layer drawn;

6) According to the document corresponding vSlide inside rtf and vcf data, as well as vText inside, the corresponding vSize and vLocation draw the corresponding text, according vSize the vImage inside the vLocation, file to draw the corresponding picture.

7) Completed first vSlide, draw, began to loop through the next vSlide draw until the end of the cycle.

4.3. VCP Editing Module

VCP presents complete, you can edit the text and pictures. The specific algorithm is as follows:

1) Get current screen vSlideId;

2) vSlideId find the corresponding vSlide, and then the vSlide to find corresponding vMaster and vBackgroud;

3) To obtain the user clicks on the position of the cursor record their points A (x, y), traverse each vText vSlide and vImage detect whether the point A belonging ([vText (the vImage). VLocation.x vText (vImage). vLocation.x + vSize.width], [vText (vImage). vLocation.y, vText (vImage). vLocation.y + vSize.heigth]);

4) If point A is a multiple vText or the vImage object whichever vLocation.z value is the corresponding vText or vImage;

5) If A is a vImage can only drag the location or size of the update, drag algorithm and zoom down the algorithm execution;

6) vMaster traversal (only the user can click Edit vMaster edit);

7) Finished traversing all vText and vImage vSlide can not be an area that contains A, then began to traverse all vText vMaster and vImage object;

8) If point A is multiple vText Or vImage object, take vLocation.z largest, to find the corresponding point A vText or vImage;

9) A belongs vText, then according to the corresponding rtf data directly edit, if you need to drag and zoom the corresponding algorithm A belongs vImage, you can only drag the position or size of the update, the corresponding drag the algorithm execution corresponding zoom corresponding algorithm;

10) vBackground traversal, if A does not belong vSlide and vMaster directly vBackground point, can not be edited.

4.4. VCP Distribution Module

The user can in the final image presented, edit, save visiting card contents. Export PDF, instant printing, and can also be shared through Bluetooth or wireless to other devices.

The **Figure 18** below is the editing effects.

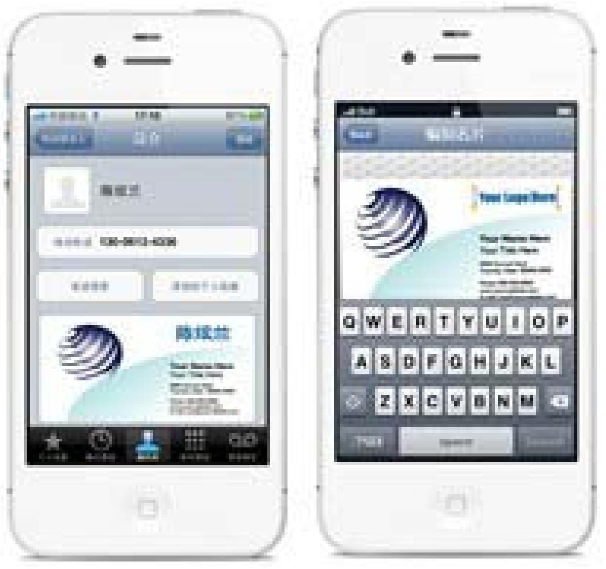

Figure 18. Editing effect chart.

5. Future Work

This paper has discussed a storage structure of the rich visiting card. As a start, it customizes the vCardPackage format, while the following steps of trans-missing private information involve the security issues. All of those (using encryption or authorization to protect the format, supporting style diversity, and renderer optimization) will take some time to accomplish.

6. Acknowledgements

This study is supported by the Basic Research Program of China/973 Project, Basic Research on Data Storage System Theory and Technology for Complex Application Environment under contract No. 2011CB302306.

REFERENCES

[1] Wikipedia, "Optical Character Recognition," 2012. http://en.wikipedia.org/wiki/Optical_character_recognition

[2] R. Smith, "An Overview of the Tesseract OCR Engine."

[3] Wikipedia, "Radio-Frequency Identification." http://en.wikipedia.org/wiki/RFID

[4] T. Kallonen, J. Vartiainen and J. Ikonen, "RFID and Visiting Card Information," *International Conference on Computer Systerms and Technologies-CompSysTech*, 2009.

[5] S. Kawanaka and H. Hosoya, "biXid: A Bidirectional Transformation Language for XML," *ICFP*'06, Oregon, 16-21 September 2006, Portland.

[6] B. Dragulescu, I. Ermalai, M. Bucos and M. Mocofan, "Using hCard and vCard for Improving Usability in Moodle," 6*th IEEE International Symposium on Applied Computational Intelligence and Informatics*, Timisoara, 19-21 May 2011.

[7] Wikipedia, "RTF." http://zh.wikipedia.org/wiki/RTF

[8] T. Howes, M. Smith and F. Dawson, "A MIME Content-Type for Directory Information," Lotus Development Corporation, The Internet Society, 1998.

[9] F. Dawson and T. Howes, "vCard MIME Directory Profile," The Internet Society, 1998.

[10] C. Jennings, J. Reschke and Greenbytes, "vCard Extensions for Instant Messaging (IM)," The IETF Trust, 2007.

[11] C. Daboo, "Extended MKCOL for Web Distributed Authoring and Versioning (WebDAV)," Apple Inc., 2009.

[12] S. Perreault, "vCard Format Specification," Internet Engineering Task Force (IETF), Viagenie, 2011. http://www.rfc-editor.org/rfc/rfc6350.txt

[13] S. Perreault, "xCard: vCard XML Representation," Internet Engineering Task Force (IETF), Viagenie, 2011. http://www.rfc-editor.org/rfc/rfc6351.txt.

[14] C. Daboo, "CardDAV: vCard Extensions to Web Distributed Authoring and Versioning (WebDAV)," Internet Engineering Task Force (IETF), Apple, 2011. http://www.rfc-editor.org/rfc/rfc6352.txt

[15] P. Saint-Andre, "vCard KIND: Application," Internet Engineering Task Force (IETF), Cisco, 2011. http://www.rfc-editor.org/rfc/rfc6473.txt

[16] K. Li and B. Leiba, "vCard Format Extensions: Place of Birth, Place and Date of Death," Huawei Technologies, Internet Engineering Task Force (IETF), 2011. http://www.rfc-editor.org/info/rfc6474.txt.

[17] D. Cauchie, B. Leiba and K. Li, "vCard Format Extensions: Representing vCard Extensions Defined by the Open Mobile Alliance (OMA) Converged Address Book (CAB) Group," France Telecom-Orange and Huawei Technologies, Internet Engineering Task Force (IETF), 2012. http://www.rfc-editor.org/info/rfc6715.txt.