

Virtual Pedigree

Genealogy Without Borders

Ryan Cheatham, Curtis Wigington, William Barrett

Introduction

Genealogists across the world face a unique and challenging problem. They attempt to sort through obscure historical data in order to find information about their ancestors, then they have to keep track of all the data they have found in an organized way. In today's information age, viewing all this data should be simple and intuitive and provide easy access to research opportunities.

Unfortunately, genealogical software applications are not simple and intuitive, and require a lot of contextual switches (a.k.a. switching between screens or views) in order to understand the data. This often causes the user to forget some data while looking for other data. As a result, the user loses context and their ability to tie things together is reduced. We aim to resolve this by implementing Elastic Paper and by building in features that allow users to find the data they need without losing context.

Abstract

Virtual Pedigree (VP) is a dynamic interface that allows a genealogist to see a pedigree chart in a smooth and simple way. Instead of placing each person statically on the screen, their position is adjusted as the chart is moved to make room for ancestors coming into the screen. This enables a quick and seamless traversal of the chart. This is done by assigning each graphic an exponential path to traverse as the user drags the chart horizontally, then adding the vertical component of the drag to the overall translation of each graphic. This and other features can be viewed at <http://youtu.be/I9FboVzmrBw>.

The development of VP has demonstrated a new dynamic way of displaying hierarchical data for both ancestors and descendants (<http://youtu.be/e0U0f2YOAB8>). Current methods of showing descendancy are static and become very large as more people are added to it. To view a large tree all at once you have to look at it from a birds-eye view which provides too much information, causing names, etc. to virtually disappear (see puzilla.org). Other dynamic methods require the tree to be redrawn, losing critical context (see FamilySearch), or require a hover-click to discover a name - both unacceptable solutions. VP allows the tree to expand and contract smoothly as the user moves through the generations while preserving context and visibility of names and dates.

The VP Dynamic Descendancy (DD) algorithm allows the descendancy tree to be dynamically (re)drawn by evenly spacing all of the descendants of a single generation as would be done in a typical static tree. However, the DD algorithm calculates and stores the relative vertical

distances between each child and their parent in the tree dynamically as the user pans and scrolls through the descendants. With these preprocessed values the VP tree is virtually somewhere between two generations. Its structure is built by interpolating between the values of two generations to make a dynamically shaped tree. Then, from one person's coordinates the rest of the tree can be built.

As the tree moves, these generation percentages change dynamically, allowing for a smooth transition from one generation tree to another without losing context. With the intuitive movement it allows for users to quickly identify color-coded missing information in the tree.

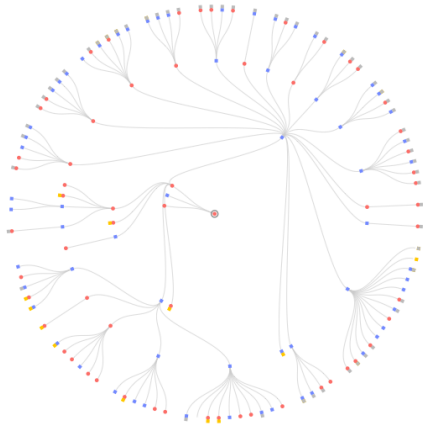


Figure 1 - Puzilla.org descendency chart

Related Work

There has been much work done to represent pedigree charts in such a way that it allows the user to effectively navigate their family history. Notable work has been accomplished in the use of fan charts to display this information. One example is puzilla.org (see **Figure 1**). This “birds-eye-view” allows the user to see a large amount of their tree at a time. However the user loses important context such as seeing names, dates, and other informations. Other pedigree charts do provide this information but only allow for a small amount of people to be seen. They are static and act as if they were paper.

Dynamic Ancestry

The core idea behind automatically adjusting the people in a pedigree chart to make room for more coming in on the screen is shown in **Figure 2**. Moving from right to left, each person in each generation fills exactly twice the vertical space as the previous generation. Thus, if each person is given an exponential path, base 2, to follow as it is moved horizontally, it will create the required space. This is the core of the algorithm that displays ancestors.

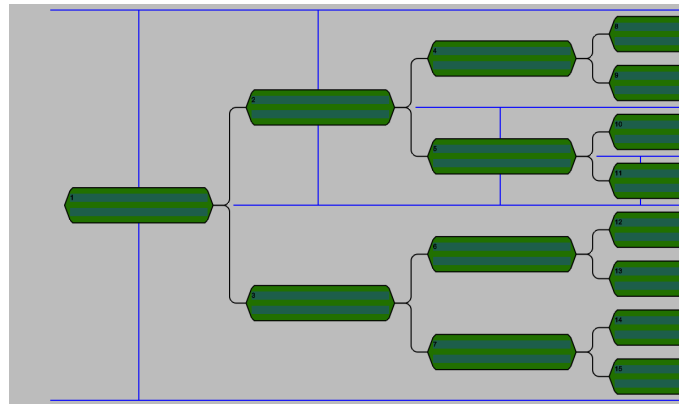


Figure 2 - Spatial requirements of Dynamic Ancestry

Dynamic Descendancy

DD.1 General Structure

The general structure behind descendancy is that at every generation there is a (blue, vertical) focus line where all person nodes are evenly spaced by a unit of one. All of the other nodes are

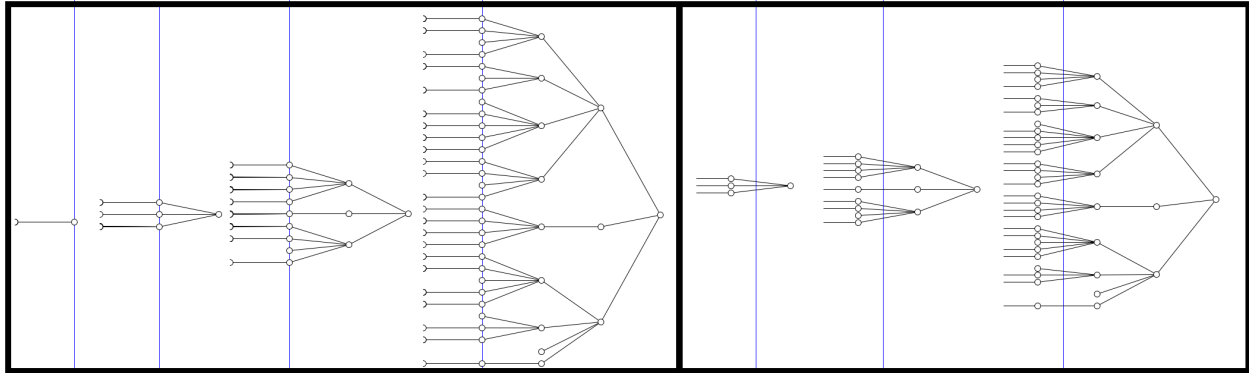


Figure 3 - Abstract Dynamic Descandancy; (a) Generation is exactly on the focus line (left); (b) Generations are halfway between the focus line (right)

positioned based on the focus line. Parents are vertically positioned halfway between their direct first and last child (see **Figure 3a**).

When the tree is drawn it is the combination of two trees built off of the focus line. They are of generations Loading... and Loading.... **Figure 3b** shows the same tree as shown in **Figure 3a** but with the states of the tree being approximately halfway between generations Loading... and Loading.... In **Figure 3b** we can see that the generations are neither evenly spaced by a unit of one nor lie on the focus line.

DD.2 Tree Relative Structure

In order to preserve the structure as described above without having to recalculate the values for every node in the tree each frame, the tree is preprocessed once. In this preprocessing there are four key values that are calculated for each node. These values are calculated at each generation for each node. These four values are Focus Child Count, Focus Children Relative Percentage, Focus Nibling Count, and Focus Nibling Relative Percentage. Nibling stands for nieces and nephews, representing the children of a node's parent who are on the focus line.

Focus Child Count (FCC): The amount of children they have at the focus line for a given generation. If they are at the generation where they are on the focus line they have 0 children on the focus. In this case we default the FCC to 1.

Focus Children Relative Percentage (FCRP): The percentage representing the node's relative position to its direct descendants on the focus. If the node is on the focus line we default the FCRP to 0.5.

Focus Nibling Count(FNC): The node's parent's

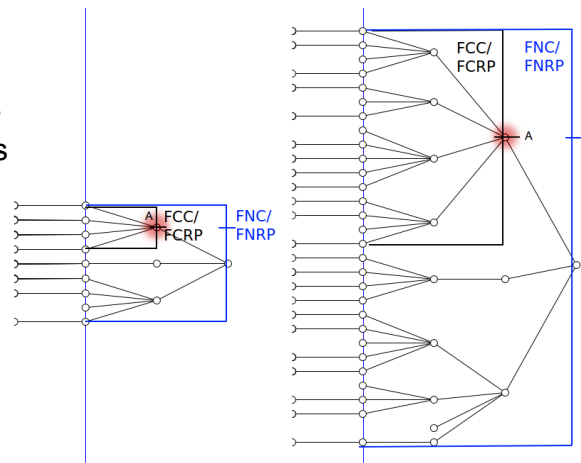


Figure 4 - Four primary values for preprocessing the tree structure

FCC.

Focus Nibling Relative Percentage (FNRP): The percentage representing the node's relative position to the node's parent's direct descendants on the focus.

Figure 4 shows a graphical representation of these values for the node represented by A. The black FCC/FCRP box shows the FCC value that represents the space occupied by the A node's children. The dash on the black box, the FCRP value, represents the node's relative position to its focus level descendants.

The blue FNC/FNRP box shows the FNC value that represents the space occupied by all of the A node's nibblings (and children). The dash on the blue line, the FNRP value, represents the A node's relative position to the focus level nibblings (and children).

DD.3 Displaying the Tree

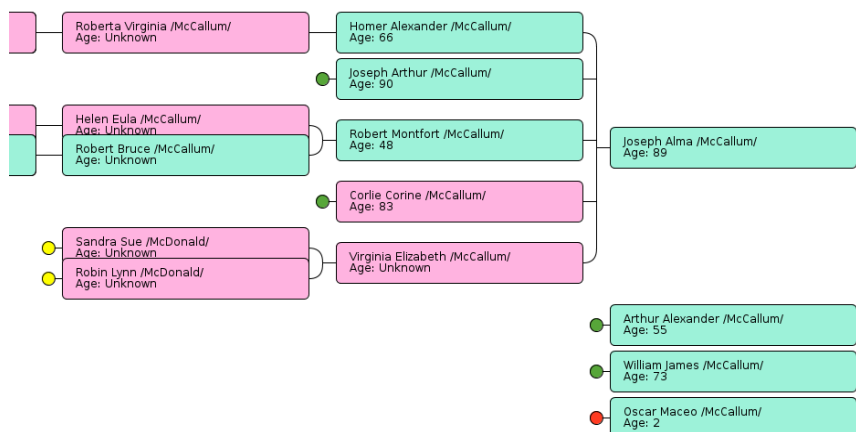
With the percentages calculated any portion of the tree can be built based on the position of any node in the tree. This is done by selecting a node and placing it at an arbitrary location on the screen. The node recursively uses the percentages previously calculated to draw all of its direct ancestors in the correct location. Then each of these ancestors draws its children. If a child has descendants that are within the viewing area, its children will recursively follow the same procedure. Whether or not the child is in the viewing area is calculated using the four values.

DD.4 Showing Real Data

The structure previously described can be represented in a more visually appealing way as shown in **Figure 5**. The colored knobs represent branches that do not continue. The knob colors can be changed to represent information such the likelihood that children exist that are not in this tree. For example, red implies no descendants (e.g. the person died as an infant). Green implies that there is a high likelihood of descendants. Yellow implies a good possibility at there are descendants. A simple polar mapping can give the same experience in the form of a fan chart.

Future Work

Currently virtual pedigree is in the early stages of development. We have only just finished the base algorithm. Our plans are to turn this idea into an application that can be easily used on many devices including phones and tablets. We plan on adding capabilities to connect with



services such as FamilySearch as to allow the user to see their family history data. Much work to improve the user's ability to identify missing information will be done as well.

Figure 5 - Dynamic Descandancy with real data