AgentVis: Visual Analysis of Agent Behavior with Hierarchical Glyphs

Dylan Rees, Robert S. Laramee, Paul Brookes, Tony D'Cruze , Gary A. Smith and Aslam Miah

Abstract—Glyphs representing complex behavior provide a useful and common means of visualizing multivariate data. However, due to their complex shape, overlapping and occlusion of glyphs is a common and prominent limitation. This limits the number of discreet data tuples that can be displayed in a given image. Using a real-world application, glyphs are used to depict agent behavior in a call center. However, many call centers feature thousands of agents. A standard approach representing thousands of agents with glyphs does not scale. To accommodate the visualization incorporating thousands of glyphs we develop clustering of overlapping glyphs into a single parent glyph. This hierarchical glyph represents the mean value of all child agent glyphs, removing overlap and reducing visual clutter. Multi-variate clustering techniques are explored and developed in collaboration with domain experts in the call center industry. We implement dynamic control of glyph clusters according to zoom level and customized distance metrics, to utilize image space with reduced overplotting and cluttering. We demonstrate our technique with examples and a usage scenario using real-world call-center data to visualize thousands of call center agents, revealing insight into their behavior and reporting feedback from expert call-center analysts.

Index Terms—Glyph, clustering, multivariate visualization

1 Introduction

Data glyphs are a common data representation that provides a means of conveying multiple data-dimensions for each entity. Data glyphs are defined by Fuchs et al. [1] as:

> "Data-driven visual entities, which make use of different visual channels to encode multiple attribute dimensions. They can be independently spatially arranged and can vary in size."

However, due to the challenges associated with scalability and the complex shapes of multi-variate glyphs, overlapping glyphs can occlude one another and become difficult to interpret, making visual comparison and analysis difficult, see Figure 1 (left). Their size constrains the number of glyphs that can be shown on a particular canvas, therefore limiting the number of data samples that can be displayed. This scalability limitation is a general perceptual and technical challenge for the use of glyphs, with large data-sets becoming more of the norm in the age of big data. Some solutions have been proposed to overcome the scalability challenge [2], [3], [4], [5], however, these techniques have not been demonstrated with a large real-world data set, multiple overlapping data samples, and a compelling industry application with big data.

The scalability limitation is especially noticeable in the call center industry, with large call centers featuring thousands of agents collectively handling hundreds of thousands of calls every day. The call center industry employs approximately 3.6 million agents in the U.S. alone [6], which signifies the importance and size of the industry to the global economy. Call centers represent the main point of contact between most customer-facing businesses and their clients. Therefore large companies require an abundant number of agents to facilitate this. It is not uncommon for large companies to employ thousands of call center agents. To provide company-wide analysis of agent behavior and performance, large scale data analysis is required with visualization of a large number of complex multi-variate data samples.

We present a novel application that incorporates and adapts a number of techniques to develop a system capable of visualizing thousands of interactive high-dimensional glyphs by utilizing aggregated clusters. We demonstrate the use of dynamic hierarchical glyphs that aggregate according to zoom level and a user-customizable distance metric to manage screen space while reducing glyph overlap. Figure 1 shows an example of our clustering on the right with the same data un-clustered on the left. The use of multiple clustering strategies developed in collaboration with a team of domain experts in the call center industry is also exhibited. We utilize these features to explore, analyze and present complex agent behavior provided by our industrial partner. Our contributions are:

- An application that explores and renders thousands of multi-variate glyphs representing real-world commercial call center data, guided by a series of feedback sessions with industry
- A novel hierarchical glyph aggregation and placement algorithm that alleviates occlusion and complexity for thousands of multi-variate glyphs
- An industry-driven usage scenario and feedback from practicing domain experts in the call center

- *Dylan Rees is with Swansea University. E-mail: 849119@swansea.ac.uk*
- *Robert S. Laramee is with The University of Nottingham. E-mail: Robert.Laramee@nottingham.ac.uk*
- *Paul Brookes and Tony D'Cruze are with QPC Ltd. E-mail: {paul.brookes, tony.d'cruze}@qpc.com*
- *Gary A. Smith is with Aria Solutions.*

Manuscript received April 19, 2019; revised March 31, 2020.

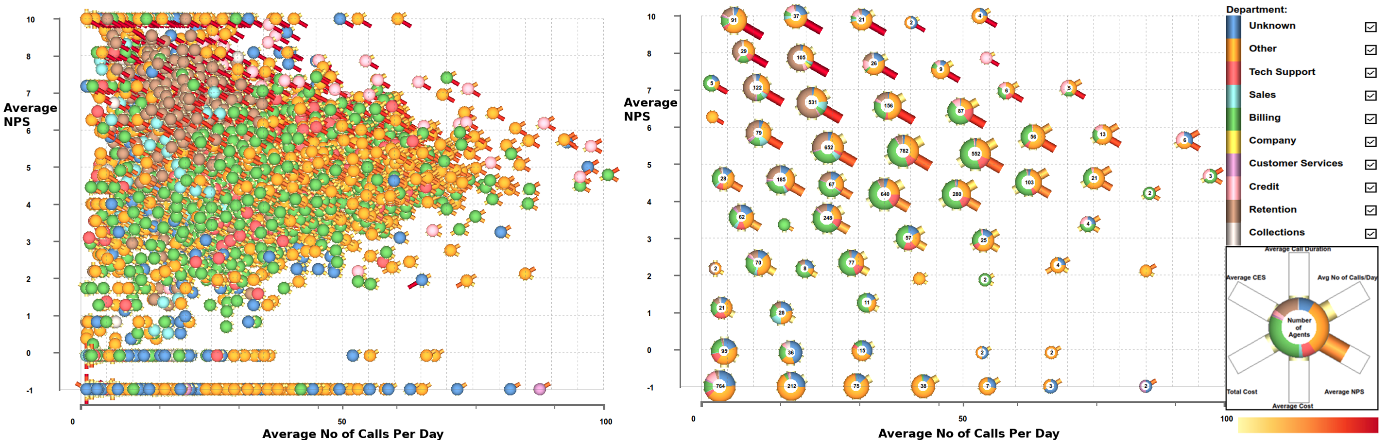

Fig. 1: The left figure shows an overplotted scatter plot with more than 6,500 glyphs. The right is the equivalent clustered hierarchical glyph with reduced overplotting and clutter. Each glyph represents a call-center agent (or a group), showing six attributes and color-mapped by their department. The x-axis is mapped to the customer feedback Net Promoter Score (NPS), and the y-axis is mapped to the average number of calls the agents take each day. The volume of agents, and the departments they represent, in each region of the plot becomes apparent after clustering has occurred with a number indicating this (see right image). We can observe that agents from the sales, company and collections departments all handle less than 50 calls per day, whereas this information was previously occluded. Outliers are also easily identified by their distance from other agents, therefore they rarely cluster.

industry

The motivation for this study was the need for our domain experts to visually explore agent behavior, something they were unable to do with currently available tools. This is the very first time our call center partners have been able to visually explore and analyze an entire month's worth of agent work with almost 5 million calls and 6,500 agents simultaneously. Individual agent behavior, represented by individual glyphs facilitate a comparison of agents, with trends and unique outlier behavior easily identifiable. Glyphs provide an intuitive form of conveying behavior data and were favored by our domain experts for this reason. They found glyphs to be intuitive for data analysts, managers, and for clients. Technical challenges associated with our development include designing and implementing dynamic glyph clustering criteria for different zoom levels, developing smooth transitions, and interactively rendering large numbers of glyphs. We also consider the influence of clustering behavior. We report on our prototype's design process utilizing guidance provided by Sedlmair et al [7]. On reflection the design process, we realize that the techniques presented in this paper are generalizable to other applications beyond studying agent behavior. The techniques and designs presented in this study are transferable to any suitable multivariate data analysis.

In the next section, we present related work on glyph design, glyph placement and call center operation with a focus on visualization. he methodology of how the collaboration with the domain experts was implemented is reported. We describe some requirements from call center domain experts and user requirements from an end user. A description of the dataset used and the attributes of interest are provided. Glyph clustering considerations and implementation are then described before an industry-driven usage scenario and domain expert feedback is reported. We finish with conclusions, future work, and reflection of the design process.

2 Related Work

Our work is related to a number of topics including glyph design, glyph placement, clutter reduction, hierarchical algorithms, and call center operation. Here we discuss related work in each of these subtopics.

2.1 Overview

McNabb and Laramee's Survey of Surveys [8] identifies three surveys related to glyphs in the information visualization domain. The first of these by Ward [9] presents a glyph placement taxonomy for multidimensional data. Borgo et al. provide guidelines on glyph design [10], and more recently Fuchs et al. reviewed a number of user-studies using glyphs to collate insights into glyph design [1]. Another glyph based survey is available in the scientific visualization field for spatial medical data [11]. For a comprehensive overview of glyph-based visualization we recommend the reader consult these surveys and for a more general overview of visualization, we recommend a survey of information visualization books by Rees and Laramee [12].

2.2 Glyph Design

A large body of work exists for different data glyph designs across multiple disciplines, from meteorological glyphs [13], through geographically mapped medical data [14], to sports visualization [15], [16]. Borgo et al. survey and collate previous work to develop design guidelines for the use of glyphs. Not all glyph designs are well received, however [17]. Sailem et al. encode up to 15 variables to a glyph for visualizing breast cancer cell phenotypes [18]. Osawa introduces 2D and 3D jigsaw-puzzle-like glyphs to show inheritance relationships in object-oriented programming [19], [20]. The main focus of our work is not the glyph design itself. We adapt a well-known multi-variate chart, a ringed bar chart, and develop a hierarchical version of it.

The same adaption could be performed with other multi-variate glyphs. Table 1 summarizes the glyphs discussed here, indicating the number of variables encoded by the glyphs and the number of data samples demonstrated. Clustering criteria are not addressed by these papers (marked by 'N/A') nor are hierarchies and clustering.

Recent work by Kammer et al. [21] utilizes a similar technique to ours by using a combination of scatterplots and glyphs for the visualization of dimensionality-reduced multivariate datasets. Users are able to pan and zoom the scatterplot, with the glyphs becoming visible by zooming, increasing the level of detail, or using a magic lens. A quad-tree aggregation is available to group data points in dense plots. Users are also able to compare different dimension reduction techniques using a side-by-side comparison. This work differs from our own as our design utilizes individually zoomable, describable axis, smooth clustering transitions, and preserves glyph visibility at all times. Our work also provides an aggregated glyph to represent cluster details, which is not implemented in the quad-tree aggregation.

2.2.1 Hierarchical Glyphs

Woodruff et al. present techniques to keep the density of data constant in a zoomable interface [2]. Techniques suggested are the aggregation of glyphs to create a single representative glyph, replacing the glyph with one of lower density and the removal of glyphs. We expand on this work with the use of multi-variate glyphs and incorporate a user-driven distance metric for clustering rather than grid based.

Yang et al. use a single star plot to represent a cluster of data points, where the values at axis intersections represent the mean of the cluster [4]. Maximum and minimum values for each variable within the cluster are also displayed as part of the glyph design. The star plot cluster is demonstrated on the Cars data set [22] of 400 data samples structured in a grid layout aggregated in 16 glyphs.

Fuchs et al. utilize transparency and aggregation techniques to cluster overlapping glyphs exploiting a leaf-based design [5]. Transparency is used to enable identification of overlapping glyphs, however, this is limited in the number of discernible overlaps. A grid is used to find clustered glyphs. Two aggregation techniques are demonstrated which they entitle Prototype Generation and Abstraction. Prototype Generation results in the creation of a new glyph of mean values, or by combining visual elements of each merging glyph to create a hybrid parent. Abstraction results in the glyphs being resized to fit around a branch representation of all glyphs in the grid cell. Fuchs et al. demonstrate their technique using Fisher's Iris data [23], and Cortez and Morais' forest fires data [24], which contain 150 and 517 data samples respectively. This work differs from ours as we develop the glyph aggregation to incorporate a dynamic zooming function on independent or coupled axes, clustering and separating glyphs according to zoom level rather than a grid. Our work also demonstrates smooth interactive zooming.

Müller et al. present multilevel glyphs for analysis of biomedical data [3]. The number of data attributes a glyph represents is dependent on the zoom level, and more data attributes are shown as the user interactively zooms in. At the highest level of detail 16 attributes are mapped to the glyphs, and at the lowest level of detail each glyph is mapped to a single color attribute, however, smooth transitions between zoom levels are highlighted as an open research question [3]. Glyph spatial positioning is determined by pre-set binned categorical attributes avoiding potential overlap of glyphs. In contrast, we always map the same number of variables to a glyph and instead aggregate the data attributes dynamically while zooming in or out. We also enable the use of continuous space for spatial positioning. Müller et al. exhibit their techniques on 70,000 data samples of cancer cases. Differences between these works and ours are summarized in Table 1.

2.2.2 Glyph Aggregation

Glyph aggregation can be achieved in a number of ways. Fuchs et al. [5] utilized a grid method, where a grid is placed on the scatterplot, and all glyphs that lay in a particular grid cell are aggregated to form a parent glyph. We consider this method limited due to the reliance of the grid placement determining the merging of glyphs. Two glyphs could potentially overlap, but have their respective centers on either side of a grid line in separate cells and therefore not merge. It is also not ideal for the representation of continuous data.

Yang et al. [4] demonstrate the use of a hierarchy tree to cluster glyphs. Clustering is based on a pre-computed hierarchy tree, with the zoom level determining the level in the hierarchy tree and therefore the number of glyphs shown. The hierarchy tree is constructed using the BIRCH method [29], however, hierarchy construction is computationally time-consuming [30]. This differs from our work as we utilize a dynamic distance clustering metric that enables interactive zooming on each axis independently, which is difficult with a single hierarchy tree.

Table 1 summarizes previous related work and compares it to ours. The work from this category most closely matches our work as a hierarchical aggregation of glyphs is utilized along with a dynamic level-of-detail utilized with the use of zooming. The work of Müller et al. [3] shows more glyphs than any other, however, the glyphs at this level represent a single variable only. Other than this, our work is demonstrated on a significantly larger number of glyphs compared to other related work. Our work is the only one to utilize a dynamically calculated distance metric to cluster agents, to provide smooth transitions between aggregation levels and to provide the ability to zoom independently on each axis.

2.3 Glyph Placement

Ward's taxonomy [9] identifies two predominant glyph placement strategies, data-driven, and structure-driven placement. In data-driven strategies, the glyph location is derived from the data, whereas structure-driven placement relies on spatial connectivity or relationship between glyphs that defines the location, such as a hierarchical relationship. Furthermore, data-driven strategies are divided into raw and derived placement. Raw data placement strategies rely on two or three data dimensions as positional components while derived data position is determined by analytic processes such as Principal Component Analysis (PCA) or Multidimensional Scaling (MDS). Our work relies on a derived

Reference	2 Number of Glyphs	3 Number of Data Variables	4 Placement Strategy	5 Dynamic Level of Detail	6 Hierarchical	7 Aggregation Method	8 Smooth Transitions	9 Independent Axis Zooming
Glyph Design								
Anderson [13]	4	5	Data Driven	N	N	N/A	N/A	N
Sailem et al. [18]	19	21	Data Driven	N	N	N/A	N	N
Chung et al. [15]	55	9	Data Driven	N	N	N/A	Y	Y
Chung et al. [16]	55	9	Data Driven	N	N	N/A	Y	Y
Tong et al. [14]	209	14	Cartographic	N	N	N/A	Y	N
Hierarchical Glyphs								
Woodruff et al. [2]	500	1	Data Driven	Y	Y	Grid Based	N	N
Yang et al. [4]	400	7	Data Driven	Y	Y	Tree Hierarchy	N/A	N
Müller et al. [3]	70,000	16	Data Driven (Categorical)	Y	N	N/A	N	N
Fuchs et al. [5]	517	14	Data Driven	N	Y	Grid Based	N/A	N
Glyph Placement								
De Leeuw and Van Wijk [25]	12	7	Manual	N	N	N/A	N	N/A
Treinish [26]	400	3	Manual + Grid	N	N	N/A	N	N/A
Ward & Lipchak [27]	96	10	Spiral Based	N	N	N/A	N	N/A
Peng et al. [28]	400	6	Grid	N	N	N/A	N	N/A
Kammer et al. [21]	Unspecified	10	Grid	Y	N	Quad-tree	N	N
AgentVis	6,500	7	Data Driven, Hierarchical	Y	Y	Dynamic Distance Driven	Y	Y

TABLE 1: A summary comparison of our work with similar related work, focused on the features implemented on our application. Indicated are the maximum number of glyphs demonstrated in the research (column 2), along with the number of glyph variables used (column 3), the glyph placement strategy (column 4), and whether glyphs provide a zoom adjustable level-of-detail (column 5). Research utilizing a hierarchical glyph is indicated (column 6), and for those applicable papers, the aggregation method is described (column 7). Work incorporating the use of smooth transitions is also indicated (column 8) as is work featuring independent axis zooming (column 9).

data-driven placement strategy using a 2D scatterplot and a customizable clustering distance metric.

Feature-driven placement is another strategy presented by Peng et al. [28], where glyphs are placed on features such as iso-surfaces. Another placement strategy is demonstrated in medical visualization [31], [32]. Other placement strategies suggested by Borgo et al. are user-driven placement, where glyphs are positioned to query data in a particular region. Examples of this include a probe for visualizing small sections of a flow field as presented by De Leeuw and Van Wijk [25] and interactive visualization of weather data by Treinish [26]. In contrast to these strategies, our placement relies on a data-driven placement strategy.

Ward also discusses distortion techniques to help avoid clutter and overlapping of glyphs. Jittering of discrete data points and shifting positions are discussed with the work of Woodruff et al. [2]

Categorical data can be binned in horizontal columns as demonstrated by Chung et al [15]. Our work plots glyphs on a continuous data scale. Glyph location is determined by continuous data values which can be distorted if using binning or jittering. Chung et al. demonstrate a single extendable axis to reduce glyph overlap on a glyph scatter plot [16]. This mimics zooming on a single axis, whereas we demonstrate zooming on both scatterplot axes and clustering of glyphs. Chung et al. also demonstrate glyph resizing for focus + context and to reduce clutter. Despite plotting a relatively small number of glyphs, occlusion is highlighted as a challenge for future work. In fact, the overlap of glyphs is one of the major inspirations behind this work.

Ward and Lipchak utilize a linear, cyclic, and spiral layout for analysis of cyclic data [27]. The cyclic layout enables easy comparison of glyphs which have a common position in a cycle, which are stacked vertically. Similarly, the spiral layout is constructed in a way that aligns common positions in the same section of the spiral. This differs from our 2D scatterplot grid layout, however, it provides an avenue for future work. See Table 1 for a summary and comparison of related work. Our system features a unique combination of thousands of glyphs, an interactive, hierarchical placement strategy, dynamic level-of-detail, smooth transitions, and independent axis zooming.

2.4 Call Center Operation

Call center operation and management is a complex subject with several strands of research including personnel scheduling, capacity planning, forecasting, and queuing. Askin et al. provide a survey of research into call center operation [33] and identify areas for future research in the operational management field.

Traditionally research into call center metrics has been statistical in nature [34] however, more recently there have been a few publications on the visualization of call center data. Roberts et al. [35] present an interactive zoomable treemap for visualizing time-dependent customer-centered call center data. Users are able to apply multivariate filters to call subsets, facilitating the discovery of patterns of customer-centered call center events and behavior. Roberts et al. demonstrate a use-case of smart brushing of parallel coordinates using a customer-centered call center database, enabling new insight into customer behavior [36]. Roberts et al. show customer-to-call center interaction through multiple call events using a visual hybrid of a state-transition diagram and a Sankey diagram [37], conveying the path of a customer journey. A GPU-assisted scatterplot application for the display of large customer-centered call center data sets is presented by Rees et al. [38] and further extended to a more feature-rich application [39]. This software enables the domain experts to visualize a large number of customer-centered calls which they were unable to do so previously. These call center visualizations all focus on the journey a customer takes through the call center whereas we present agent-centered analysis visualizations. Our work is also the only call-center based work which utilizes the multi-variate nature of glyphs for high-dimensional visualizations and explicit clustering.

3 Methodology

In this section, we outline how we collaborated with the domain experts to design our system. Three domain experts were initially consulted for this project, each focusing on different aspects of the call center industry (programmer, researcher, analyst). The first expert is a lead software developer who has 30 years of expertise in the call center industry. Expert two is a consultant with experience in a variety of roles within the industry, totaling over 20 years. The third expert has 15 years of call center experience and is a director of product and marketing. He is also an Associate of the Operational Research Society. These three experts are employed at QPC Ltd. Data for this project was provided by QPC Ltd. for the purpose of development. A description of the data is provided in Section 4.

Analyzing the domain requirements: This project is the latest in a long collaboration spanning over five years and numerous projects [35], [36], [37], [39]. Through this time over 10 formal meetings have been conducted, with eight meetings dedicated specifically to this project. These meetings were held with all three domain experts. Meetings were recorded and archived for later evaluation and reference. Through these discussions requirements of the analysts were established and validated in subsequent meetings. These requirements are outlined in Section 5.

Prototyping visual designs: During discussions, visual designs were constructed using a whiteboard and the advantages and disadvantages of each design were discussed before a design was agreed to be implemented. Input was gathered from all three domain experts. Periodic meetings were held to make sure that the implementations matched the expectations of all three domain experts.

Feedback gathering: Once the implementation had been complete, feedback was gathered over a number of guided interview sessions. Interviews were semi-structured according to guidance provided by Hogan et al [40]. Feedback was initially gathered from the same three domain experts, however, one of the collaborating domain experts has since departed QPC Ltd. and was no longer able to provide further feedback. See Section 7.6 for a discussion on this. Additional feedback was also gathered from a front-line call-center agent.

4 Agent Call-Center Data Description

Our industry partner QPC Ltd. provides the data utilized to develop this application. QPC Ltd. provides infrastructure and data logging for call centers. As such they have access to extensive data requiring analysis to provide insight and to inform efficiencies of call centers.

The dataset provided comprises of almost 5 million calls collected over a month for one of their clients. These calls were collected across multiple sites in multiple countries representing calls with over 6,500 agents. These agents work in 10 call center departments as follows (List 1):

- Tech Support
- Sales
- Billing
- Company
- Customer Services
- Credit
- Retention
- Collections
- Unknown
- Other

Each agent has several attributes for analysis, compiled from all calls that an agent is involved with. These attributes (or characteristics) are as follows (List 2):

- Average NPS
- Average Cost
- Total Cost
- Average CES
- Number of Transfers
- Percentage of Transfers
- Average Call Duration
- Number of Calls
- Average Number of Calls per Day
- Number of Short Calls
- Percentage of Short Calls
- Average Hold Duration
- Number of Stand Alone Holds
- Number of Consult Holds
- Number of Transfer Holds
- Number of Cold Transfer Holds
- Number of Warm Transfer Holds
- Number of Conference Holds
- Total Hold Count

Of particular interest to the experts is the NPS or Net Promoter Score. This is a feedback score provided by customers who contact the call centers and complete a survey. Customers are asked to rate their experience of their call center contact on a scale from 0 to 10, with 0 being negative and 10 being positive. Few calls receive an NPS score, approximately 1% of calls, however it represents the ground truth measure of customer satisfaction (actually reported by customers in surveys) and can be used as an indicator of operations improvement. Those which do not provide a feedback score are assigned a -1 score.

Customer Effort Score (CES) is another attribute of interest. This variable provides a derived value quantifying the effort made by the customer in contacting the call center. For example, if a customer is kept waiting a long time and passed through multiple agents, this would incur a large CES, whereas a customer on a short call, speaking to a single agent would acquire a low CES. The CES is a multifaceted metric derived from a number of variables.

Hold events have been separated into the reason for the hold. Of interest, in particular, is the cold and warm transfer holds. Cold transfer holds indicate when customers are placed on hold for a transfer to another agent, and information about the call is not transferred between the agents. In contrast, warm transfer holds are for customers put on hold for agent transfers where information about the call is transferred.

5 User Requirements & Design Goals

This application is developed in collaboration with domain experts from the call-center industry who tasked us with developing a visual tool that enables insight into the data that they collect. In particular, experts are interested in call-center agent behavior. We have been working with our domain experts since 2014 on call-center visual analytics. For this particular project, focusing on call center agent behavior, design requirements were drawn up in one of our meetings with three of the domain experts. The following requirements were established to drive the development of the application:

- **R1** An application that enables exploration of multi-variate call-center agent behavior.
- **R2** The capacity to handle thousands of agent representatives.
- **R3** The ability to simultaneously compare multiple agent attributes
- **R4** An interactive interface that allows the user to probe for information and read raw data.

Typical tasks were identified throughout the consultation with the domain experts:

- **T1** Enable exploration, investigation, and comparison of agent performance.
- **T2** Enable the identification of agent behavior outside of the norm – finding outliers.
- **T3** The ability to compare groups of agents from different business operations categories such as department, site, and skill.

The visualization community also provides us with guidelines for creating visual designs, in particular, Shneiderman's Visual Information-Seeking Mantra [41] of overview first, zoom and filter, then details on demand. Due to the requirement to be able to visualize a large number of agents, clutter is an additional consideration. Ellis and Dix provide a taxonomy of clutter reduction techniques [42], and Ward's taxonomy contains considerations for glyph based designs [9], in particular, they highlight the need for smooth transitions. By taking these into consideration we develop the following design tasks to guide our development:

- **D1** Develop an application for multi-variate agent exploration, analysis, and visualization.
- **D2** Provide a visual design with an initial overview displaying all agents.
- **D3** Utilize cluttering reduction techniques to provide clutter-free imagery.
- **D4** Enable exploration with flexible zooming and smooth transitions.
- **D5** Supply agent filtering options.
- **D6** Provide both cluster-based and individual agent details on demand.

Currently, analysis of agent based information is performed manually utilizing spreadsheet tools such as Microsoft Excel.

The requirements and tasks established are not unique and are common in most multi-dimensional data analysis projects. Therefore the techniques developed here are transferable to many other domains.

5.1 Alternative Design Solutions

A number of potential designs were discussed before the solution we implemented was agreed upon. We identify a number of design solutions from consultation of a multi-dimensional visualization survey by Liu et al. [43] and other works [44], [45], [46].The survey also discusses multi-dimensional scaling techniques [47], although we chose not to develop these as the abstraction required was deemed nonintuitive and non-accessible to clients.

Multiple axis solutions such as parallel coordinates [48] and an Andrews Plot [49] are other potential solutions. A parallel coordinate solution is implemented for another related project, as part of the same collaboration [36]. However, challenges with scalability were identified including overplotting and with the comparison of agent attributes across multiple axes. Although there are solutions to the overplotting challenge [50], [51], [52], simultaneously comparing multiple attributes across multiple axes is difficult (**T1**). A crowded Andrews Plot also suffers the same problem, with tracking a single line across multiple axes difficult, therefore this design solution was also discarded.

A scatterplot matrix [53] is an alternative method of multivariate visualization that we use to provide an overview of the dataset (**D2**). However tracking a single agent across multiple plots is again difficult and clutter is again a challenge with scatterplots.

Multiple linked views present another design solution. Although we have not implemented this solution, it provides a complementary solution that can be added to our implementation. Coordinated multiple views are part of the future work for this project.

A design consisting of glyphs in a scatterplot layout was chosen as glyphs convey a number of variables for a particular agent, enabling the comparison of multiple attributes simultaneously. Glyphs on a scatterplot are also very intuitive. This accessibility is a requirement in order for the domain experts to communicate with clients. Unlike other design choices such as parallel coordinates or scatterplot matrices, glyphs provide a clarity of identification of which variables belong to which agent, glyphs group the visual cues of each agent making it easier to see all the attributes associated with the agent together, rather than trying to track polylines across multiple axes, or points across multiple plots. Our use of glyphs on a scatterplot matrix has challenges associated with scalability, although we try to alleviate this problem with a glyph clustering technique.

6 Hierarchical, Agent-Based Glyphs

6.1 Workflow Overview

The application is designed for the following general workflow: Firstly a scatterplot matrix of all available agent variables is presented once a dataset has been loaded (Figure 2). Each plot provides an indication of variable correlation, see Section 6.2. The user is able to select a plot from the matrix to show those agent attributes as the main axes in the focus plot. Here each point represents an agent as a multivariate glyph. For the next step, to reduce clutter, the user may choose to cluster overlapping glyphs, as can be seen in Figure 1 and described in Section 6.3. Additional clustering options are available by individual teams, departments, agent skill and call center site.

Filtering is an optional step that allows for further clutter reduction (Section 6.5). Exploration is the next step, the user is able to explore the space by zooming and panning the scene, with glyph clusters updating dynamically. This is exemplified in the the supplementary video [54], [55]. For further exploration, users are able to customize the glyphs using a variety of designs and with agent attributes of their choice (Figure 3).

Finally, for identifying the agent represented by the glyph, hovering over an individual glyph provides details of the agent associated with the glyph as can be seen in Figures 5 & 7.

To meet the requirements and tasks outlined in Section 5 we developed an application following the guidelines of Shneiderman's Visual Information-Seeking Mantra featuring a scatterplot matrix overview (**R1**) for initial exploration of the data and to find correlating variables. To enable simultaneous comparison of multiple agent attributes we developed a multi-variate glyph based visual design (**R1**, **R3**), with each glyph visualizing the characteristics of a single call-center agent. Glyphs are placed in a 2D scatterplot with axes mapping determined by agent properties (**T1**). We chose a scatterplot in order to facilitate comparison of agents (**T1**). Scatterplots are a familiar and popular form of visualization, Lee et al. preform an extensive study and highlight scatterplots amongst the twelve most popular visual designs [56], while the use of glyphs facilitate the simultaneous comparison of multiple attributes (**R3**). Scatterplots were also chosen as a result of our discussion with QPC Ltd. They are familiar with scatterplots and can easily convey the results of exploration and analysis to their clients and managers. Scatterplots enable the sorting of agents based on any of their characteristics. This capacity enables fast identification of both outlier (**T2**) and average agent behavior as we shall demonstrate (see Figures 1,5,6,7). With thousands of agents to display, clutter is a significant challenge. To counteract this, we cluster glyphs into hierarchical parent glyphs, removing overlap (**T3**, **D3**). Users are able to interact with the glyphs by zooming in independently on each axis (**R4**, **D4**), with clusters updated dynamically to minimize clutter and occlusion. See the supplementary video for a demonstration of the application [54]. We developed the software in Qt, C++ using OpenGL for the implementation.

6.2 Overview Scatterplot Matrix

To provide an overview of the data-set (**D2**), we provide a scatterplot matrix of all agent characteristics. This can be seen in Figure 2. A correlation coefficient is also calculated for every agent variable combination. This correlation coefficient is overlaid on each of the grid of the scatterplot matrix, facilitating the search for correlating variables. To further enhance the overview, this correlation coefficient is also used to colormap the background of each scatterplot in

Fig. 2: A scatterplot matrix of all agent variables. A value for the correlation coefficient can be seen on each of the individual scatterplots. Background color is also used to indicate the correlation coefficient. Of the variables shown (scrollbars can be used to view more agent characteristics) the Number of Transfers and Total Cost show the highest correlation, identifiable by the darker red background color. Plots along the diagonal of the scatterplot matrix show a perfect correlation coefficient of 1.0 as they show the same variable on both axes. These are automatically given a gray background to indicate their irrelevance.

the matrix. Scatterplots with a higher correlation coefficient display a dark red background color, with lower correlation having a lighter red background, as in Figure 2. The scatterplot matrix can be accessed using a UI button, bringing up the view (as seen in Figure 2) in a new window. Clicking the plot of interest places that plot into the focus view. An option allows for the matrix window to be kept open after clicking, for use in a split screen/dual monitor scenario, otherwise the window will close.

6.3 Glyph Design and Hierarchicalization

The hierarchical glyph can be derived from a number of pre-existing designs, however, a new visual encoding channel is dedicated to showing the number of glyphs represented by the parent glyph as a result of aggregation. We demonstrate the application with a number of glyph designs developed in conjunction with our industry partner in the following sub-sections. The focus of this work is not to create a single design solution, but rather to present the aggregation method and to outline the process of hierarchicalization of a glyph to show the aggregation. Borgo et al. and Fuchs et al. both provide guidelines for glyph design [1], [10] however they do not present guidelines for creating an aggregated, hierarchicalized glyph.

Singular glyphs: Our application features a selection of glyph designs, each of which capable of displaying six agent attributes (**R1**, **D1**), plus their department. These designs include a bar chart, a combination of an ellipse with a bar chart, and a radar glyph design. See Figure 3 for a comparison of these glyphs. An example of a single agent, child glyph can be seen in the left of Figure 3, a combination of an ellipse with a bar chart with the six bar lengths representing a chosen call center agent attribute. These bars also feature a color redundantly mapped to the value of a call center agent to facilitate visual comparison of agent behavior (**T1**). The maximum extent of the bars indicate the maximum value of the mapped attribute found in the dataset. The glyph center of a single agent glyph is color-mapped to a categorical variable. This increases the analysis potential and aids differentiation between agents. In our call center agent data set this is mapped to agent call center department, enabling comparison of relationships across departments (**T3**).

In our initial multi-variate glyph design process, we investigated a range of literature to find some of the most popular ones [8]. The bar and radar chart are very popular. In addition we designed a custom glyph (Figure 3, top) in conjunction with the same three domain experts from QPC Ltd. After discussing the full range of possibilities from the literature with QPC, and then providing implementations of the bar, radar, and elliptical glyphs, we focused on the elliptical design due to its symmetry and ease of legibility on a scatterplot. Chung et al. went through a very similar glyph design process for a different application, but also employing a scatterplot layout [15], [16]. They also arrived at an elliptical glyph design, ours is comparable to theirs. This is one inspiration behind our work.

Hierarchical glyphs: Once two or more glyphs are clustered, a new parent glyph is created as can be seen in the right of Figure 3. Hierarchical glyphs support clutter reduction and minimization of occlusion (**D3**). They also support comparison of groups of agents (**T3**). The process of glyph hierachicalization is to start with a child glyph and to add additional encoding channels for the number of child glyphs represented, as well as an indication of the proportion of agents from each department. Parent attribute values shown in the glyph are updated to represent the mean value of all child glyphs. It is important that a parent glyph is distinguishable from a child glyph so that they are not misinterpreted. The parent glyph for the elliptical glyph with bars is shown in Figure 3. The parent glyph differs from the child glyph with the addition of a center ellipse and an indicator of the number of child glyphs in the cluster. The colored segments forming the ring around the glyph center are also updated to construct a donut-like chart indicating the proportion of each agent's call center department represented by the glyph. The protruding bars are also updated to show the mean values of all children for each agent attribute.

In general the process of glyph hierarchicalization is fairly straightforward. To create a multi-variate parent glyph, a number is added to the glyph center reflecting the number of children it represents. The size of the glyph itself is also optionally mapped (logarithmically in our case) to the number of children represented. Thirdly, the average value of the child attribute is reflected in the parent, consistent with the child glyph. The parent does not strictly have to represent an average of its children, other group properties could be reflected such as maximum attribute values. This is not part of the requirements in this application however.

Although not implemented in any glyph design within this paper, other meta-data from the clustered child glyphs can also be encoded within the parent glyph design. As an example, Yang et al. encode maximum and minimum values as bands surrounding the mean values of aggregated child glyphs in a star glyph [4]. Other meta-data that can be encoded include the standard deviations.

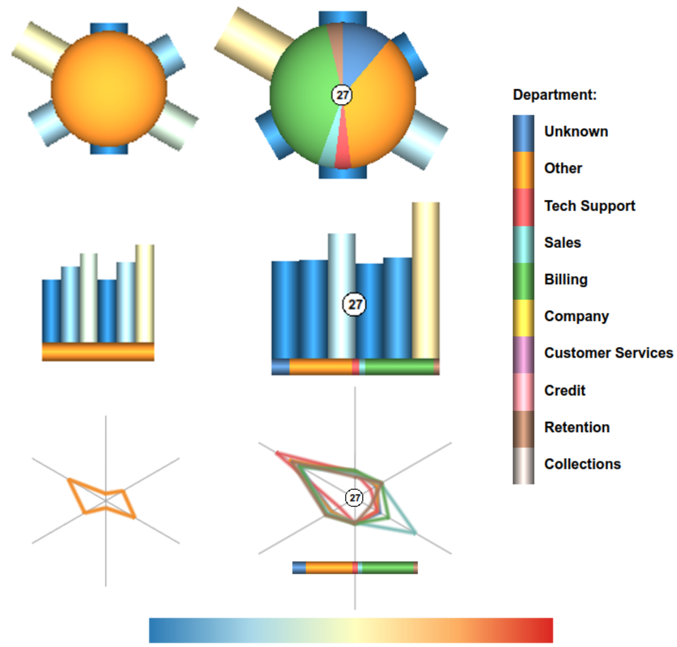

Fig. 3: Three designs of child (left) and parent glyphs (right). The top glyph features six bars, or spokes, surrounding a central ellipse, each of which represents an agent property. The center ellipse is color-mapped to a department, on the left child glyph, representing a single agent, and a single department, whilst on the parent glyph, segment size represents the proportion of each department constituted by the group of agents. Parent glyphs also feature a number at the center to indicate the number of child agents represented by the glyph. The middle glyph features a bar chart design with six agent properties, as well as a bar underneath the glyph to indicate the department and department proportions in the child and parent glyph respectively. The bottom glyphs are based on a radar design, again with six agent attributes. The child glyph features a colored line to indicate the department while the parent glyph features an average line for each of the departments represented within the glyph. A bar underneath the parent glyph indicates the department proportions. The legend to the right indicates the department color-mapping of the glyph centers. The bottom color legend is mapped to any agent attribute other than department, e.g. average hold time.

Glyph Legend: The glyph legend indicates a property of agent behavior mapped to each of the bars protruding outwards from the glyph, along with an indication of the maximum size of each bar (Figure 4 right). The maximum extent of each bar is calculated from the largest value of the represented attribute from all agents. The colormap applied to the bars are also visible below the glyph. The colormap and length of bars is redundant in this design. A separate legend indicates the categorical colormap for the center of the glyph. The glyph legend is also used to display the mean values of the entire collection of call center agents for each mapped attribute (**D2**).

Colormaps: Based on feedback and discussion with the domain experts, the user has a choice from a selection of

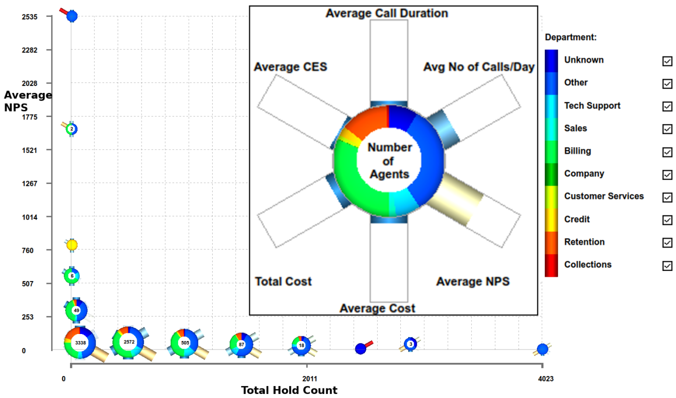

Fig. 4: Clusters plotted on a chart with total hold count along the x-axis and agent CES on the y-axis for over 6,500 agents (**R2**). Glyph bars below average have been removed, leaving only bars with above-average values. This highlights some interesting findings, the bar indicating average agent NPS, at the 4 o'clock position, is only visible on glyphs with a low CES and for agents with a low total hold count. Agents with a low total hold count also have a low total cost, as indicated by the absence of the total cost bar at the 8 o'clock position from those glyphs.

colormaps to customize aesthetics. By clicking either the department color legend, to the right of the plot, or the glyph bar color legend, below the glyph legend, a choice of color maps are shown for the user to select. Colormaps were chosen from tested and established sources. Categorical colormaps for glyph centers are collected from Colorgorical [57], Tableau [58], and Telea [59]. Color maps for the data bars are based on sequential and diverging colormaps from ColorBrewer [60]. Various examples of colormaps can be seen in figures throughout this paper. The application is designed to support any color map requested by a user (**D1**). The default colormaps used based on the Tableau 10 colormap and the Cool to Warm colormap developed by Moreland [61]. The Tableau 10 colormap is the default colormap used in Tableau and was built around the human perception influenced CIELAB colour space [62], while the Moreland colormap is the default colormap used in Paraview [63].

6.4 Glyph Legend Interaction

Interaction options are provided for glyphs to facilitate exploration, analysis, and selection of agent attributes (**D5**). By clicking an individual bar on the glyph legend, the selected bar on all glyphs in the scatterplot toggles between focus and context, as demonstrated in Figure 8. This option renders the selected bar with less opacity, removing the focus from that bar. User selection enables highlighting individual glyph bars and agent variables. A sub-set selection option is also provided that displays all glyph bars with either above or below average agent values for a given agent attribute (**T1**, **D5**). See Figure 4. This option applies to all bars across all glyphs. By utilizing this option with the call-center agent dataset, above or below average agent behavior is easily observable.

Clicking the center of the glyph legend returns a dialog from where agent property mapping of each of the glyph bar can be customized allowing the user to tailor the agent properties that can be seen and compared. Any of the attributes in List 2 on page 5 can be individually chosen. By default glyph variables are attributes indicated by the domain experts as most important. These are the variables that can be seen in Figure 4.

6.5 Filtering and Selection

At the request of the domain experts, we provide a filtering feature based on user-chosen departments, enabling focus to be placed on individual call center departments (**T3**, **D5**). This focus + context filter removes glyphs according to the categorical variable to which the color is mapped. With the call-center dataset, we have mapped the agent's call center department to this attribute. This is due to the importance of being able to compare agent behavior based on department. Each agent department has a checkbox next to the color map on the legend as can be seen in Figure 5. Selecting these places each glyph belonging to that category into focus or context. When a glyph is rendered as context it is displayed as a semi-transparent gray disk in the background and is not considered for clustering. If a particular category is filtered while already clustered, all parent glyphs are recalculated with the exclusion of the filtered category. An example of this selection and filtering can be seen in Figure 5 which plots the NPS against the number of cold transfer calls. An interesting outlier agent who has a low feedback score and a high number of cold transfers is highlighted with the on-mouse-over dialog. The glyph bar at the 8 o'clock position shows that a high number of warm transfers are also attributed to this agent. The on-mouse-over dialog indicates a large number of short calls and a very large number of holds, suggesting that these factors have contributed to the agent receiving a low NPS metric.

Glyph Rendering Options: The user can interactively adjust glyph scale. The user also has the option to map glyph size to the number of clustered agents represented by the given parent glyph (**D4**). Due to the potentially large number of agents clustered into a single glyph, the size scale increases logarithmically with the number represented. Shading is added to the glyphs, these effects have been used for improved aesthetics and to provide a 3D effect. This is achieved by utilizing OpenGL lighting effects and varying normals across the glyphs.

Detail Options: To facilitate data exploration, on-mouse-over information is available for each glyph providing original data values for that particular glyph (**D6**, **R4**). On clustered glyphs, the number of agents from each department represented in the cluster is provided. Figures 5 (bottom) and 7 show examples of the dynamic tooltips feature.

6.6 An Agent Distance Metric for Clustering

To determine if two glyphs should join to form a cluster, the x and y-axis component distances are tested along with the current zoom level of each axis to determine the Euclidean distance between each glyph in screen space (**D3**). If this is

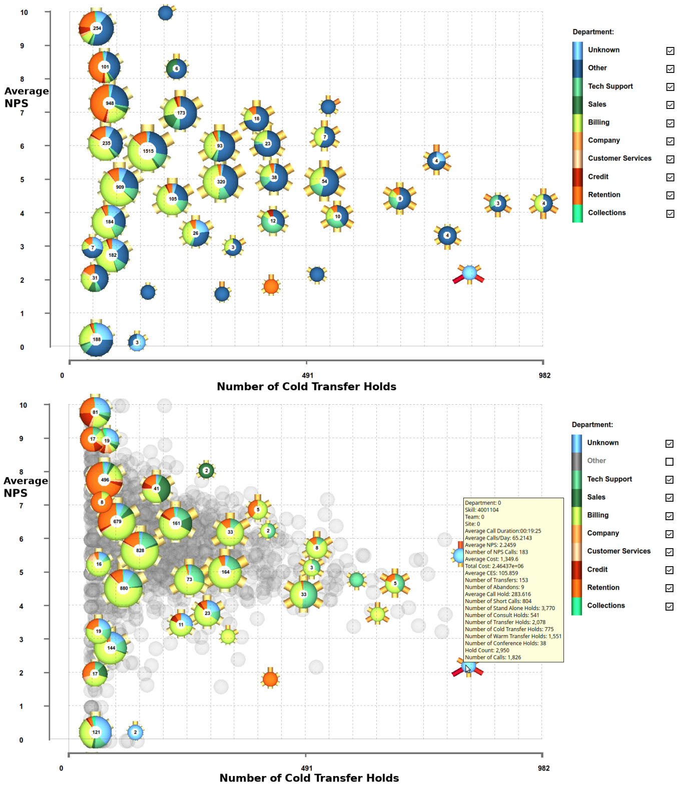

Fig. 5: Filtering of agent department. Both figures show the agent NPS on the y-axis against the number of cold transfer calls on the x-axis. The top image shows all departments while the bottom image shows the same scene with agents from other departments filtered and rendered as a gray context in the background, see Section 6.5. Mouse-over interaction is shown in the bottom figure, highlighting an interesting, outlier agent with a low NPS score and a very high number of warm transfers indicated by the bar on the 8 o'clock position of the glyph. The on-mouse-over dialog also indicates a large number of short calls, in this case, calls less than five seconds, and a very large number of holds, suggesting that these factors contribute to a low NPS score.

below the clustering distance (Equation 1) the glyphs are merged. Our customized clustering distance is as follows:

$$d(a) = s_x \frac{d_x}{d_{max}} + s_y \frac{d_y}{d_{max}} + s_z d_\alpha \tag{1}$$

The components of the distance metric are:

- Euclidean distance (d_x): This constituent measures distance between two agents a_a, a_b in screen space using $d_x = |a_a^x - a_b^x|$, where s_x is a scaling factor in the x-direction controlled by the user.
- Euclidean distance (d_y): This measures distance between two agents a_a, a_b in screen space along y using $d_x = |a_a^y - a_b^y|$, where s_y is a scaling factor in the y-direction controlled by the user.
- Agent attribute (d_α): This constituent compares agent attributes such as department, skill, or team between two agents a_a, a_b where $s_z = \infty$ IFF $a_a^\alpha \neq a_b^\alpha$ i.e. the user turns on clustering by department otherwise $s_z = 0$, $\alpha \in \{department, site, team, skill\}$

The user is able to control the distance, $d(a)$, at which clustering occurs via a slider in the interface. This enables control of the number of visible glyphs in the screen space and the amount of glyph overlap which is dependent on glyphs size, also adjustable by the user. By default, the clustering distance is set so that a balance is achieved between the overlap and the glyph size. We also implement other clustering controls incorporating feedback from the domain experts. Clustering can be customized to only occur if the agents share a common property, α. The user has a selection of categorical properties to choose from for this clustering constraint, in the case of the call center agent, these are the agent department, site, skill level, and team. This enables the domain experts to concentrate on and compare specific categories of agents (**T3**). Examples of this can be seen in Figure 6, where agents are grouped by department, showing the average values for each call-center department. Observable are the three distinct clusters of departments. The Sales and billing department cluster have almost double the hold duration in comparison to the credit, company, and collections department cluster, suggesting that effort could be made to reduce hold time in the sales and billing departments. Figure 7 shows a plot with clustering constrained to those within the same department, highlighting agents with different behavior in comparison to others from the same department. A customer services agent has been highlighted with a high number of calls per day and a low percentage of short calls, unlike any other from the same department. Short calls are calls that last less than 5 seconds, too short a time to be productive, generally indicating undesirable behavior from the agent.

6.7 Details vs Abstraction

Due to the nature of big-data, overplotting is a challenge, common to most visual designs. The use of clustering in this work is our solution to this challenge, however, as with most solutions, a compromise between detail and data abstraction is made. The more glyphs that are clustered and the data aggregated, the more abstract the glyphs become. The abstraction is linked to the number of child glyphs. As the user zooms in, the fewer glyphs in the screen space and therefore, the less abstract the representation. Overplotting becomes a greater challenge as datasets become larger, therefore, the clustered glyphs also become more abstract with growing datasets.

Glyph overlap is observable in Figure 7 (top) due to a constraint of glyphs only being able to cluster with those representing agents from the same department. However, by zooming in to the region indicated by the red box, these overlaps can be eliminated, with the resultant plot shown in Figure 7 (bottom). The zooming enables more detail to be seen, and reduces the overlap and abstraction.

6.8 Clustering Operations and Placement

Glyph placement is driven by user chosen agent attributes on both the x-axis and y-axis, described by Ward as raw data-driven placement [9]. Either continuous and non-continuous data can be mapped to both axes, which generally leads to overlapping of glyphs. By enabling interactive zooming of the scene, overlapping can be reduced as axes

Fig. 6: A plot of average CES against average hold duration with agents grouped by departments. Notable are the three very distinct clusters of departments, with the company, credit, and retention forming a cluster with a low hold duration and CES, sales and billing in another high hold duration and CES, and the other departments lie in a cluster in between. See Section 6.6. Additionally, this Figure shows the application layout and controls. The insert shows additional clustering controls from the hidden 'Clustering' tab.

expand when zooming in. If axes variables are the same for overlapping glyphs, this will have little effect. Another method to redress overlapping is by changing opacity as described by Fuchs et al. [5], this is however limited by the number of glyphs that can be stacked before perception challenges arise. Alternatively, overlapping can be resolved by clustering glyphs into a single hierarchical glyph representation. This clustering method is equivalent to testing Euclidean distance between glyphs and merging glyphs within a pre-defined distance, Equation 1. This distance metric is interactively customized by the user (**R2**, **D3**). Once glyphs are clustered, the positioning of the new parent glyph is determined by the mean of agent-based axis variables of all child glyphs. We call this data-driven hierarchical placement.

The glyph ordering when calculating Euclidean distance influences the result of the clustering. A glyph equidistant between two others could cluster to either of the other two, depending on which one was tested first, giving different clustering results. To ensure a constant clustering order, it is therefore important to ensure that glyph order and priority is consistent. To address this, the user has the option to sort glyph priority by any agent attribute in List 2.

6.9 Cluster Aggregation Options

Cluster aggregation can be achieved in a number of ways. Fuchs et al. [5] previously utilized a grid method, while Yang et al. [4] demonstrate the use of a hierarchy tree to cluster glyphs. We discuss different strategies that dynamically test glyph distances at each zoom level and enable zooming on each axis independently. This is because agents may overlap heavily along one dimension only (and not the second one). This gives the user greater control and enables easier exploration of the data (**R1**, **D1**). This is exemplified in Figure 8, where the user has zoomed the x-axis, average hold duration, yet kept the full range of the y-axis, average

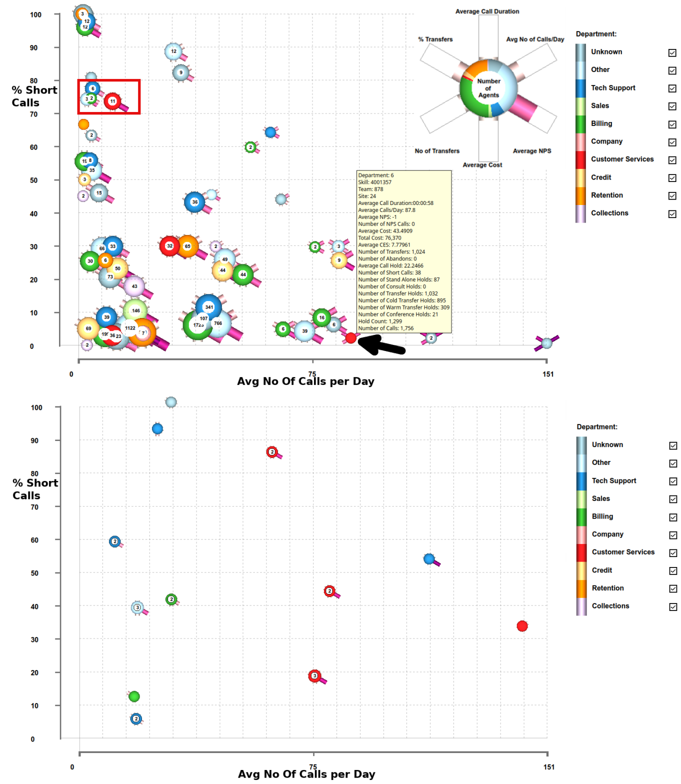

Fig. 7: (Top) A plot showing the percentage of short calls against the average number of calls per day. Clustering is specified to within departments only. Although this can result in some overlap, it enables outliers within departments to be easily discovered. A customer services agent is highlighted with an arrow and the on-mouse-over as such an example. No other agent from this department has the same combination of a high number of calls per day and a low percentage of short calls. (Bottom) A zoomed in view of the red highlighted section from the top, showing glyphs dispersed after zooming.

NPS, to see the full distribution of NPS scores between a particular hold duration. Observable is the distribution of agent departments, with the retention department in sky-blue most prevalent at the high feedback score, while agents from the billing department are predominantly found with a lower score. We can also observe the inverse trend between average NPS score and average hold duration for the billing department. However there is an outlier billing agent with a high NPS score and longer than average hold duration, indicated with an arrow. This agent may deserve recognition.

Dynamic Hierarchical Glyphs: Dynamically hierarchical glyphs are interactively recalculated at each zoom level or triggered by a user modification to the cluster distance metric. Parent glyphs are discarded at each frame, and new ones recomputed. This entails testing each glyph against every other glyph against the distance metric $d(a)$. The Pseudo-code 1 demonstrates this. The convention we use for pseudo-code is:
A lowercase $'g'$ represents a child glyph, e.g. g_t,
A lowercase, boldface '**g**' represents a list of child glyphs, e.g. $\mathbf{g}_{list}$,

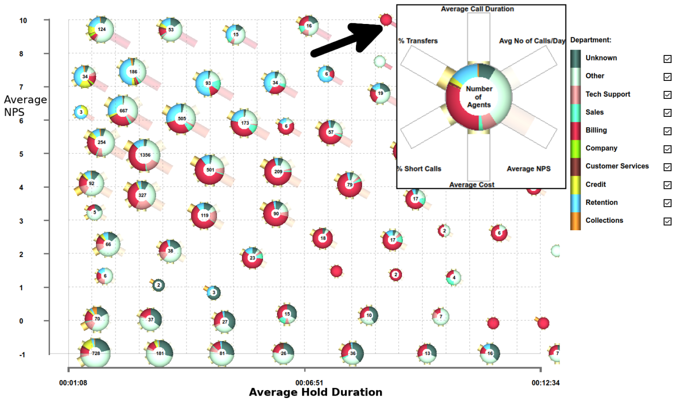

Fig. 8: A figure displaying the agent NPS score on the y-axis and the average hold duration on the x-axis. The user has zoomed the x-axis to a hold duration between 1 minute and 12.5 minutes while keeping the y-axis at the maximum extents to explore the full NPS distribution. The user has also filtered the 2 and 5 o'clock bars, average NPS and number of calls per day, from the glyph. We discover that the agents belonging to the billing department, in red, are primarily concentrated below an NPS score of 7, with few agents achieving a higher customer satisfaction score. In contrast, agents representing the retention department, in sky-blue, primarily have an NPS of over 5, with few agents receiving a low customer satisfaction score. An outlier agent is indicated with an arrow. See Section 6.9

An uppercase $'G'$ represents a parent glyph, e.g. G_t,
An uppercase, boldface $'\mathbf{G}'$ represents a list of parent glyphs, e.g. $\mathbf{G}_{list}$.

While using this technique, although it is fast, it was found that it could be confusing to the user as parent glyphs disappear and new parent glyphs reappear when the scene changes. Thus, in order to increase cohesiveness, we developed the next version that preserves parent glyphs between frames.

Dynamic Hierarchical Glyphs with Parent Preservation: To alleviate the confusing experience from the dynamically created glyphs, we modify the algorithm to include the memory of parent glyphs between frames. Initial merges are calculated as previously outlined for dynamically created glyphs to create a list of parent glyphs. On subsequent scene changes, parent glyphs are preserved. Instead, each child glyph within each parent glyph is re-tested based on the current distance, $d(a)$. Glyphs above the distance are individually removed from the parent glyph. If a parent glyph is left with a single child, the parent glyph is replaced by the child.

Once this has been updated, each parent glyph is tested against every un-merged glyph, and every other parent glyph, with glyphs being added to the parent if they fall within the threshold distance $d(a)$. Finally, all un-merged glyphs are tested against each other to potentially create new parent glyphs. Using this algorithm, we are able to support interactive exploration of a dataset containing over 6,500 agents. The process is outlined in Pseudo-code 2.

Yang et al. previously utilized a pre-computed binary hi-

```
Input Variables:
g_list        List of all agent glyphs
d             Glyph distance threshold, Eqn 1
Local Variables:
g_t           Current glyph being tested
g_c           Candidate glyph for clustering
g_clustered   List of glyphs already clustered
g_tested      List of glyphs already tested
g_parent      List of glyphs clustered to a parent
Output Variables:
G_list   List of parent glyphs

FOR EACH g_t ∈ g_list //for each child glyph
    IF g_t ∉ g_clustered //if glyph is un-clustered
        FOR EACH glyph g_c ∈ g_list
            IF g_c ∉ g_tested AND g_c ∉ g_clustered
                IF |g_t − g_c| < d
                    g_parent.add(g_c)
                    g_clustered.add(g_c)
                END IF
            END IF //g_c ∉ g_tested AND g_c ∉ g_clustered
        END FOR
    END IF //g_t ∉ g_clustered
    g_tested.add(g_t)
    IF g_parent NOT ∅
        g_parent.add(g_t)
        G_list.add(g_parent)
    END IF
    g_parent.clear()
END FOR
```

Pseudo-code 1: Dynamic hierarchical glyph clustering

erarchy tree [4] to determine what glyphs are shown at each zoom level. However, as we provide independent zooming on each axis, a single hierarchy tree is more difficult. A hierarchy tree would be required for every zoom point on an axis, this would theoretically be an uncountable set of zoom levels, however, a fixed number of resolution zoom points could be set. To pre-compute all possible trees in a large dataset would require a large amount of computing resources and would be impractical, thus we have not implemented such a solution.

Smooth, Interactive Transitions: To enhance the perception of two glyphs joining together and clustering, an animated transition is implemented that lets the user follow the paths of child glyphs to a parent cluster, (**R2**, **D4**). This prevents a visual disconnect that arises if glyphs disappear and parents appear incoherently. An animation is also used for splitting of parent glyphs for the same reasons. The animation is achieved by dynamically calculating the position of a clustered glyph and using this as a target location for the children that are merging at that time. Glyph positions are linearly interpolated and updated to support dynamic motion this target over a short time. Figure 9 shows an image of transitions in motion. To view the smooth animated transitions, see the accompanying **video** [54].

7 Evaluation

We evaluate the application by utilizing its features to investigate the relationship between CES and call duration. We learn that this varies by department, suggesting an allowance be made in the computation of CES for the department. Further observations are available in the accompanying **video** [54]. Following the usage scenario, we report feedback from domain experts.

```
Input Variables:
$\mathbf{g}_{list}$        List of all agent glyphs
$\mathbf{g}_{clustered}$   List of glyphs already clustered
$\mathbf{G}_{list}$        List of parent glyphs
$\mathbf{G}_{clustered}$   List of parent glyphs
$d$                        Glyph merging distance, (Eqn 1)
Local Variables:
$g_t$                      Current glyph being tested
$g_c$                      Candidate glyph for clustering
$G_t$                      Current parent glyph being tested
$G_c$                      Candidate parent glyph for clustering
$\mathbf{g}_{tested}$      List of glyphs already tested
$\mathbf{G}_{tested}$      List of parent glyphs already tested
$\mathbf{g}_{parent}$      List of glyphs clustered to a parent
Output Variables:
$\mathbf{g}_{list}$        Updated list of all agent glyphs
$\mathbf{g}_{clustered}$   Updated list of glyphs already clustered
$\mathbf{G}_{list}$        Updated list of parent glyphs
$\mathbf{G}_{clustered}$   Updated list of parent glyphs

FOR EACH $G_t$ ∈ $\mathbf{G}_{list}$ // Remove children too far away
    FOR EACH glyph $g_t$ ∈ $G_t$
        IF |$g_t$ − $G_t$| > $d$
            $G_t$.remove($g_t$)
            $\mathbf{G}_{clustered}$.remove($g_t$)
    END FOR
END EACH
FOR EACH $G_t$ ∈ $\mathbf{G}_{list}$ // Test for parents and children
                                    // to cluster
    IF $G_t$ ∉ $\mathbf{G}_{clustered}$
        FOR EACH $G_c$ ∈ $\mathbf{G}_{list}$
            IF $G_c$ ∉ $\mathbf{G}_{tested}$
                IF $G_t$ ∉ $\mathbf{G}_{clustered}$
                    IF |$G_t$ − $G_c$| < $d$
                        $G_t$.add($G_c$)
                        $\mathbf{G}_{clustered}$.add($G_c$)
                    END IF
                END IF
            END IF
        END FOR
        FOR EACH glyph $g_t$ ∈ $\mathbf{G}_{list}$
            IF $g_t$ ∉ $\mathbf{G}_{clustered}$
                IF |$g_t$ − $G_t$| < $d$
                    $G_t$.add($g_t$)
                    $\mathbf{G}_{clustered}$.add($g_t$)
                END IF
            END IF
        END FOR
    END IF
    $\mathbf{G}_{tested}$.add($G_t$)
END FOR
FOR EACH $g_t$ ∈ $\mathbf{G}_{list}$ //Test for new parents
    IF $g_t$ ∉ $\mathbf{G}_{clustered}$
        FOR EACH $g_c$ ∈ $\mathbf{G}_{list}$
            IF $g_c$ ∉ $\mathbf{G}_{tested}$ AND $g_c$ ∉ $\mathbf{G}_{clustered}$
                IF |$g_t$ − $g_c$| < $d$
                    $\mathbf{G}_{parent}$.add($g_c$)
                    $\mathbf{G}_{clustered}$.add($g_c$)
                END IF
            END IF
        END FOR
    END IF
    $\mathbf{G}_{tested}$.add($g_t$)
    IF $\mathbf{G}_{parent}$ NOT ∅
        $\mathbf{G}_{parent}$.add($g_t$)
        $\mathbf{G}_{list}$.add($\mathbf{G}_{parent}$)
        END IF
    $\mathbf{G}_{parent}$.clear()
END FOR
```

Pseudo-code 2: Dynamic hierarchical glyphs with parent preservation clustering

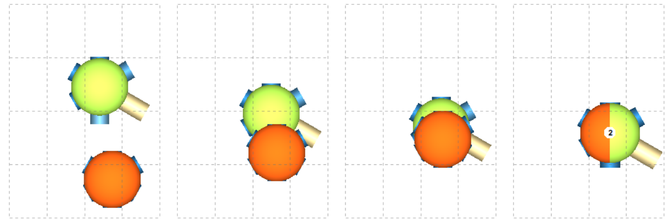

Fig. 9: A figure displaying four frames from a smooth animated transition of agent glyphs, representing different departments, following user zooming. See the accompanying video for animation [54]

7.1 Evaluation Justification

Our evaluation comprises of an industry-inspired usage scenario and reporting of feedback from domain experts. We deem this to be an appropriate evaluation, as this application is built for special-purpose domain experts in the call center analytics industry, and not more general users. This also reflects guidelines set out by Munzner [64]. Although an improved evaluation could be achieved with a long term field study, this could be a direction for future work.

7.2 Usage scenario – Customer Effort Score Validation

In addition to examples of new knowledge described previously, an industry-driven usage scenario is provided to demonstrate the utility of the application.

The usage scenario is a demonstration of a workflow that an analyst may apply while using the application. The typical workflow for the application follows Shneiderman's Visual Information-Seeking Mantra of overview first, zoom and filter, then details on demand [41]. With the application this involves first utilizing the scatterplot matrix to find correlating variables and to select the focus scatterplot axis variables. Other data variables of interest can then be mapped to the glyphs to facilitate exploration of the data. The clustering, along with the filtering and zooming provides clutter reduction. Finally the mouse-over feature displays individual agent details. The findings demonstrated in this usage scenario represent real knowledge derived from the use of the AgentVis application whilst exploring the data.

Identified as of particular interest by our domain experts is Customer Effort Score (CES), a metric derived to asses the amount of effort the customer made in engaging with the call center. As the score is a derived value, our industry partners are interested in assessing the validity of the score, particularly in relation to how the score varies across different departments and how this compared to the customer feedback Net Promoter Score (NPS), considered as a ground truth indication of customer satisfaction from surveys. We demonstrate a typical workflow to investigate how the CES and NPS correlate, inspired by the domain experts and the tasks that they would typically preform.

To investigate this, a dataset incorporating a month's worth of calls, close to 5 million, is used. We first selected the average agent CES on the y-axis, and place the average call duration on the x-axis (**R1**). This leads to a very cluttered view so we, therefore, utilize the cluster feature for ease of exploration (**D3**). This leads to a single parent glyph based in

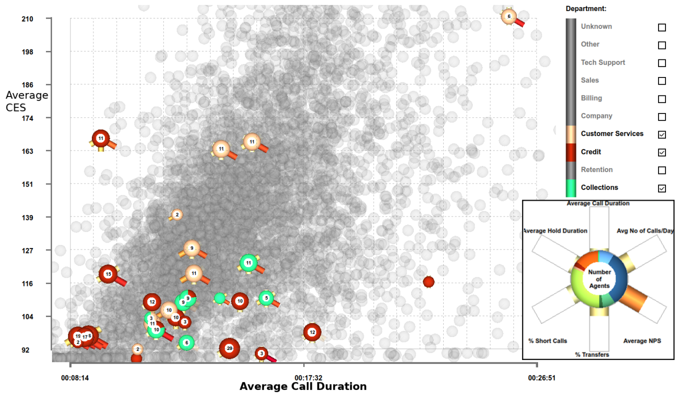

Fig. 10: A figure from a usage scenario investigating the effectiveness of CES. Average agent CES is on the y-axis and the average call duration on the x-axis. Agents have been filtered to only display those from the credit, collections, and customer services departments, and have been clustered into their teams. See Section 7.2 for more information.

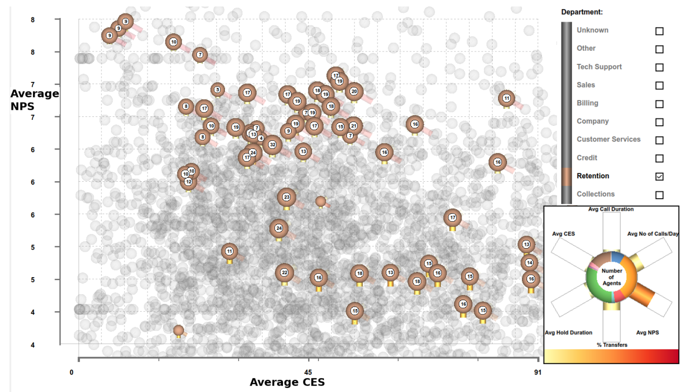

Fig. 11: A figure from a usage scenario investigating the effectiveness of CES. Average agent CES is mapped to the y-axis and the average call duration on the x-axis. Agents have been filtered to display only those from the retention department, and have been clustered into their teams. The 6 o'clock bar on the glyph is selected to show the percentage of calls transferred and the other bars are rendered as context with lower opacity. This allows the user to observe that the teams that have a lower NPS also tend to have a higher percentage of call transfers. See Section 7.2 for more information.

the lower left of the scatterplot representing the vast number of agents, with a few other clusters and single agents spread across the graph. This indicates that the distribution of the data is in the lower values of each axis, thus we zoom each axis independently to focus on the main body of the data (**R4**).

We select the option that only clusters agents if they share a common department to investigate the differences between departments (**T3**) and adjust the clustering distance metric to find a balance between screen space utilization and clutter (**R2**, **D4**). To reduce clutter further we filter departments to only show agents from the credit, collections, and customer services departments as these departments are of particular interest to the user.

As we want also to investigate how the CES compares with NPS, we ensure that the NPS is mapped to one of the glyph bars, along with other variables of interest (**R3**). We select the option to filter bars that show below average variables. The resulting plot can be seen in Figure 10.

From the image, we learn that average call duration is a driver of CES, as expected, with a general trend seen in the filtered agents rendered in context, and in the focus agents. We observe that in agents from the customer services department, the call duration has a greater effect on the CES compared to agents from the collections department who increase at a shallower incline. Agents from the credit department, however, do not seem to follow this trend with no obvious correlation between call duration and CES.

When comparing the NPS, mapped to the 5 o'clock position on the glyphs in Figure 10, we observe that agents from the credit department generally have higher values. No apparent correlation is evident between either axis and the NPS value. Also evident in the figure is the presence of outlier agents such as the lone credit agent, with longer call duration in relation to other credit agents (**T2**). This agent is also shown to have a low NPS compared to other agents (**T1**).

To investigate the relationship between NPS and CES for each department further, we map the agent NPS on the y-axis and the agent CES on the x-axis. We hypothesize that call duration, hold duration and the number of transfers have the most significant effect on customer satisfaction. Therefore we ensure that the average call duration, the average hold duration, and the percentage of calls transferred are all mapped to glyph bars.

We filter through different individual departments to search for correlation. To reduce clutter, we cluster agents by their teams. From the resultant sales department plot, we can see a strong negative correlation between the NPS and CES, with an increase in CES returning a decrease in NPS. From comparing glyph variables, we notice that a relationship also exists between the bar mapped to the agent hold duration. It also shows a strong negative correlation to the NPS, however, this is not the case for the percentage of transferred calls and call duration. This tells us that for the sales department, the hold duration is particularly important.

Similarly, a negative correlation is observed between the NPS and the CES for the retention department. Also observable for the retention department is a correlation between the NPS score and the glyph bar representing the percentage of calls transferred, where teams with a large percentage of transfers tend to have a higher CES and a lower NPS (see Figure 10). This informs us of the importance of the number of transfers for the retention department. These examples are shown in the accompanying video [54].

From this usage scenario, we can conclude that average call duration is a driver of the CES, however special consideration should be given to the department being contacted. It is possible to conclude that different departments have particular properties that influence customer satisfaction NPS.

7.3 Domain Expert Feedback

To evaluate the application, we conducted demonstrations and interviews with QPC Ltd., the data suppliers, throughout the development of the software. Their input guided development and provided regular feedback on the application. Here we provide some of their feedback.

A demonstration of the software was first conducted to introduce the clustering concept used, and to exhibit the interaction options available to the agent, highlighting some agent insights found during development.

This was the first time any of the experts have ever seen interactive exploration of 6,500 call center agents simultaneously. During the demonstration, calls had been clustered into parents with individual agents left scattered, expert three observed that it was easy to find unusual agents (**T2**): *"It's good for finding outliers in this sort of view."* In particular, an outlier agent was discovered that was not clustered. The identified agent had particularly poor performance with a low feed back score and where 30 calls out of 31 made were less than 10 seconds in duration, too short to achieve anything useful.

The experts gave positive feedback for the scatterplot-matrix view, and the overview it allowed onto the data (**D2**). Expert three commented: *"I like the high level view that gives"* Expert one, meanwhile, noticed the correlation indication commenting: *"There's a lot of things here that you'd expect to correlate... Yeah, it's good, I think it's a good initial view... The ability to click on it and go view, filter to that dataset is really good."*

The experts were explicitly asked for their opinion of the available glyph designs, to which expert one replied: *"I think, of the three, the donut one is definitely the best for me the others I'm not as fussed on, but I think it's just good to have options... I'm sure other people have different preferences."*

When exploring individual agents, expert two appreciated the on-mouse-over function (**R4**), allowing him to see all of the call details of a standout agent: *"That agent you hovered over, more than half of their calls are less than 10 seconds... that isn't enough time to establish a need to transfer. That's an interesting one."*

After demonstrating the ability to filter call center departments, and to cluster agents based solely on their team, skill or site (**T1**, **T3**), expert one commented: *"I like the fact you can select the departments now, that's great, that's really useful because you're going to get a similar pattern of calls for all the agents within that one department. The different departments are going to have different patterns, so to be able to look at one or even compare two or all of them at the same time. That's great."*

When shown the clustering function reducing the clutter on a plot with over 6,500 agents, expert two showed his approval: *"That's a good idea. I like that. It makes it more practical."* After a few minutes of exploration, expert two then declared: *"It's all very interesting, you can definitely see some patterns in there."*

Following the demonstration, structured questions were asked to ascertain the expert's opinions of the application. When asked if they could see a benefit of for the software, expert two replied: *"Yeah, I can. So given that, on the drop downs on the left you can change the axes and change the different values on the glyph on the different clock positions, there's a wide variety of things you could do with this."* This indicates the expert's appreciation of the software's ability to display multiple agent attributes (**R3**). Expert two continued: *"The whole thing looks really good to me, and it seems really flexible. It looks like it does an awful lot. I really like it."* Underlining the expert's enthusiasm for the application and its abilities (**R1**).

Expert one appreciated the application's ability to visualize a dataset comprising of all calls for an entire month, with thousands of agent represented (**R2**), something that they had previously been unable to achieve: *"It really does let you look at a big dataset in a way that we, well personally what I've used in the past, hasn't been able to do. We can definitely find stuff that I didn't know before."* Expert one also provided a spontaneous comment for the aesthetics of the application: *"I think it's very visually appealing as well, it looks unique, it looks different to the usual sorts of visualizations you get, it does grab your attention that it's different."*

Feedback was garnered for subsequent versions of the application, with expert two keen to enable exploration of features not available in the current dataset: *"The dataset that you've got within this is not as rich as some of the datasets that we've got, unfortunately, so you are limited in what you can possibly display because the underlying data isn't in the format you need it."* Ensuring that the richer datasets are compatible with the software and that new properties and variables are utilized is future work for the project.

Additional feedback was gathered from a front-line call-center agent, to provide a different perspective on the software. This domain expert has over two years experience as a call-center agent working within a sales and customer services departments at Admiral Group plc. Feedback was again garnered from a guided interview following guidelines by Hogan et al [40]. The domain expert provided feedback in a two hour session, initiated with a demonstration of the software.

On being shown the initial scatterplots the agent noted some agent behavior reflected in the data. *"The scatterplot is useful for identifying agents that are only taking a very low number of calls. This suggests that they are unavailable."* Availability of agents is very important to operations.

The zooming of the scatterplots was also praised, *"I think it's very good because then you can see what each agent is doing."* and *"I like that because it's so easy to understand."*

While exploring the data, the agent was noticing behaviors that occurs within the call center, while exploring short calls he noted, *"Calls could be shortened due to end of shift."* This new hypothesis requires follow-up investigation to be confirmed. Another insight provided was relating to customer behavior when contacting the call center, *"The sales department line is usually free. However, the customer service line tends to charge a fee. So you'll have customers calling the sales line in order to speak to the customer service department."*

The agent was asked for his overall thoughts about the software and how it would supplement his current workflow, *"To supplement with this would be perfect. It shows you exactly what you are doing, where you can improve, not only that, different departments different sites, for competition purposes because of different call centers, a lot of them use incentivization of performance, so you do have all that data, you can see that. It's quite visible." "The clustering by department is definitely useful. Companies want to see which departments and sites are* *performing the best." "It is useful... it has to be very easy to understand (for managers and analysts) and very quick with minimal cost in time."*

7.4 Conclusions and Future Work

In this paper, we utilize multivariate glyphs in a scatterplot layout to visualize call center agent behavior. We demonstrate an application with a month's worth of call center data, representing 6,500 agents handling nearly 5 million calls. To address clutter we cluster glyphs using a customized clustering algorithm based on Euclidean distance, and extensions of this to cluster agents by their department, site, skill, or team. This represents a novel solution to the problem of overplotting for glyph based layouts and we provide guidelines for the hierarchicalization of glyphs. We report domain expert feedback on the application created to visualize the data, gathered from several interview sessions.

Some limitations of using the dynamically created glyphs with memory have been noted. These are that glyphs may be different when zooming in compared to when zooming out, potentially leading to confusion. Modifying the algorithm to provide consistent glyphs at different zoom levels is future work.

Developing a workable hierarchical tree solution that allows for zooming on independent axes is another avenue for future work. Further incorporating new variables from richer call datasets is another desirable outcome.

A question also remains in relation to how the hierarchical glyphs are interpreted, especially as the user interacts with the view by zooming, although feedback from our domain experts suggest that it is intuitive. The use of animation of the children glyphs clustering into the parent glyphs is implemented to aid perception, however its effectiveness has not been fully explored. To further explore this, we propose a user study to fully understand how the clustering is interpreted.

A more thorough evaluation in the form of a long term field study, once software has been deployed and used in the workflow, would provide a more complete study of the effectiveness of the design.

7.5 Application Generalization

The techniques developed and presented in this study can be generalized to other domains and datasets. Many of the requirements established in Section 5 are not unique to call-center agent behavior domain with large multivariate datasets being ever-more prevalent. The glyph clustering algorithm is applicable to other glyph designs and datasets, as are the interaction techniques. The application specifically developed for this study, however, requires some modifications in order to being used in other domains due to customization of the given data format and domain-specific attributes.

7.6 Study Evaluation

This study led to the creation of a tool for the exploration of call-center agent behavior. The tool has since been supplied to the collaborating company for testing. However the tool has not been incorporated into their daily workflow due to limited developer resources to commercialize it or perform field evaluations. This challenge is a pitfall of a design study collaboration, highlighted by Sedlmair et al. [7], of insufficient time available from collaborators (PF-5).

A reflection of the experience on this project with the other pitfalls indicated by Sedlmair et al. reveals that most pitfalls were avoided and some new ones encountered. Challenges were, however, encountered with the task not being regularly preformed preventing a full validation of the application (PF-10).

Other challenges identified from the list identified by Sedlmair et al. where improvements could be made is with the reporting of the domain expert feedback, where the experts involved with the design also provided feedback (PF-26). Although not explicitly objective, feedback was balanced with criticisms as well as praise for the design received. Another potential pitfall suffered by his study is the lack of a long term user study before reporting (PF-32). This is as a result of PF-5 as previously discussed.

A new pitfall this project encountered was the departure of a collaborating domain expert towards the end of the project. This pitfall isn't explicitly highlighted by Sedlmair et al., therefore we recommend adding this as the 33rd pitfall PF-33. Since the departure was near the end of the project, an additional expert was found to only provide feedback. This pitfall would have had a greater impact had it happened earlier, however, disruption was minimized.

Feature creep is another new challenge encountered [65], where domain experts would request many additional features once a prototype had been seen, expecting a tool that is able to preform additional tasks. We identify this as a new pitfall (PF-34). Some requests were easily implemented, such as UI adjustments. Other requests were discussed and a triage process was carried out to determine a priority list of features. An example of a requested feature not implemented, was for a mechanism to show customer calls handled by a user-chosen agent, as halo glyphs encircling the agent glyph.

Linked to feature creep is the expectation of features being added without considering the development time required for each feature. We call this the expectations pitfall (PF-35). For the customer halo glyph request, this feature would require substantial development with additional data analysis, further customer metrics, a separate glyph design, and interface mechanisms. To temper these expectations, we recommend discussing the development investment requirements for such features, as was done in this case.

Although not necessarily a pitfall, a minor boundary to the collaboration was the physical distance between the researchers and domain experts. With a 3–4 hour traveling distance, collaboration was not as fluid as it could have potentially been, despite the use of conference calls. Had the collaborators been more physically accessible a closer collaboration could have been conducted.

Acknowledgments

The authors would like to thank Beryl Rees for her help and support during this project.

Undeb Ewropeaidd European Union Llywodraeth Cymru Welsh Government Cronfa Gymdeithasol Ewrop European Social Fund

The authors gratefully acknowledge funding from KESS. Knowledge Economy Skills Scholarships (KESS) is a pan-Wales higher level skills initiative led by Bangor University on behalf of the HE sector in Wales. It is part-funded by the Welsh Government's European Social Fund (ESF) convergence programme for West Wales and the Valleys.

REFERENCES

[1] J. Fuchs, P. Isenberg, A. Bezerianos, and D. Keim, "A systematic review of experimental studies on data glyphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1863–1879, 2017.

[2] A. Woodruff, J. Landay, and M. Stonebraker, "Constant density visualizations of non-uniform distributions of data," in *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '98. New York, NY, USA: ACM, 1998, pp. 19–28.

[3] H. Müller, R. Reihs, K. Zatloukal, and A. Holzinger, "Analysis of biomedical data with multilevel glyphs," *BMC Bioinformatics*, vol. 15, no. 6, p. S5, May 2014.

[4] J. Yang, M. O. Ward, and E. A. Rundensteiner, "Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets," *Computers & Graphics*, vol. 27, no. 2, pp. 265 – 283, 2003.

[5] J. Fuchs, D. Jäckle, N. Weiler, and T. Schreck, "Leaf glyphs: Story telling and data analysis using environmental data glyph metaphors," in *Computer Vision, Imaging and Computer Graphics Theory and Applications*, J. Braz, J. Pettré, P. Richard, A. Kerren, L. Linsen, S. Battiato, and F. Imai, Eds. Cham: Springer International Publishing, 2016, pp. 123–143.

[6] ContactBabel, "US contact centres: 2019-2023 the state of the industry & technology penetration," ContactBabel, Tech. Rep., 2019.

[7] M. Sedlmair, M. Meyer, and T. Munzner, "Design study methodology: Reflections from the trenches and the stacks," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 12, pp. 2431–2440, 2012.

[8] L. McNabb and R. S. Laramee, "Survey of surveys (sos) - mapping the landscape of survey papers in information visualization," *Computer Graphics Forum*, vol. 36, no. 3, pp. 589–617, 2017.

[9] M. O. Ward, "A taxonomy of glyph placement strategies for multidimensional data visualization," *Information Visualization*, vol. 1, no. 3-4, pp. 194–210, 2002.

[10] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-based visualization: Foundations, design guidelines, techniques and applications." in *Eurographics (STARs)*, 2013, pp. 39–63.

[11] T. Ropinski, S. Oeltze, and B. Preim, "Survey of glyph-based visualization techniques for spatial multivariate medical data," *Computers & Graphics*, vol. 35, no. 2, pp. 392–401, 2011.

[12] D. Rees and R. S. Laramee, "A survey of information visualization books," *Computer Graphics Forum*, vol. 38, no. 1, pp. 610–646, 2019.

[13] E. Anderson, "A semigraphical method for the analysis of complex problems," *Proceedings of the National Academy of Sciences*, vol. 43, no. 10, pp. 923–927, 1957.

[14] C. Tong, L. McNabb, R. S. Laramee, J. Lyons, A. Walters, D. Berridge, and D. Thayer, "Time-oriented Cartographic Treemaps for Visualization of Public Healthcare Data," in *Computer Graphics and Visual Computing (CGVC)*, T. R. Wan and F. Vidal, Eds. The Eurographics Association, 2017.

[15] D. H. Chung, P. A. Legg, M. L. Parry, R. Bown, I. W. Griffiths, R. S. Laramee, and M. Chen, "Glyph sorting: Interactive visualization for multi-dimensional data," *Information Visualization*, vol. 14, no. 1, pp. 76–90, 2015.

[16] D. H. S. Chung, M. L. Parry, I. W. Griffiths, R. S. Laramee, R. Bown, P. A. Legg, and M. Chen, "Knowledge-assisted ranking: A visual analytic application for sports event data," *IEEE Computer Graphics and Applications*, vol. 36, no. 3, pp. 72–82, May 2016.

[17] M. Correll, "Ross-chernoff glyphs or: How do we kill bad ideas in visualization?" in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '18. New York, NY, USA: ACM, 2018, pp. alt05:1–alt05:10.

[18] H. Z. Sailem, J. E. Sero, and C. Bakal, "Visualizing cellular imaging data using phenoplot," *Nature Communications*, vol. 6, p. 5825, January 2015.

[19] N. Osawa, "Visualization of inheritance relationships using glyphs," *IEICE Transactions on Information and Systems*, vol. 85, no. 1, pp. 275–282, January 2002.

[20] ——, "Application of jigsaw-puzzle-like 3d glyph to visualizing grammatical constraints," in *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*, Sep. 2005, pp. 293–295.

[21] D. Kammer, M. Keck, T. Gründer, A. Maasch, T. Thom, M. Kleinsteuber, and R. Groh, "Glyphboard: Visual exploration of high-dimensional data combining glyphs with dimensionality reduction," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[22] E. Ramos and D. Donoho, "Cars Dataset," http://davis.wpi.edu/xmdv/datasets/cars.html, uRL Accessed: 2019-02-27.

[23] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[24] P. Cortez and A. d. J. R. Morais, "A Data Mining Approach to Predict Forest Fires using Meteorological Data," in *New Trends in Artificial Intelligence, 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence*, J. Neves, M. F. Santos, and J. Machado, Eds. Guimarães, Portugal: APPIA, Dec. 2007, pp. 512–523.

[25] W. C. de Leeuw and J. J. van Wijk, "A probe for local flow field visualization," in *Proceedings of the 4th Conference on Visualization '93*, ser. VIS '93. Washington, DC, USA: IEEE Computer Society, 1993, pp. 39–45.

[26] L. A. Treinish, "Task-specific visualization design," *IEEE Computer Graphics and Applications*, vol. 19, no. 5, pp. 72–77, Sep. 1999.

[27] M. O. Ward and B. N. Lipchak, "A visualization tool for exploratory analysis of cyclic multivariate data," *Metrika*, vol. 51, no. 1, pp. 27–37, Jul 2000.

[28] Z. Peng, R. S. Laramee, O. Deussen, D. Keim, and D. Saupe, "Vector glyphs for surfaces: A fast and simple glyph placement algorithm for adaptive resolution meshes." in *Vision, Modeling, and Visualization*, 2008, pp. 61–70.

[29] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: An efficient data clustering method for very large databases," *SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, Jun. 1996.

[30] P. Berkhin, *A Survey of Clustering Data Mining Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 25–71.

[31] T. Ropinski, M. Specht, J. Meyer-Spradow, K. Hinrichs, and B. Preim, "Surface glyphs for visualizing multimodal volume data," in *VMV*, 01 2007, pp. 3–12.

[32] J. Meyer-Spradow, L. Stegger, C. Döring, T. Ropinski, and K. Hinrichs, "Glyph-based spect visualization for the diagnosis of coronary artery disease," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1499–1506, Nov 2008.

[33] Z. Aksin, M. Armony, and V. Mehrotra, "The modern call center: A multi-disciplinary perspective on operations management research," *Production and Operations Management*, vol. 16, no. 6, pp. 665–688, 2007.

[34] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao, "Statistical analysis of a telephone call center," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 36–50, 2005.

[35] R. Roberts, C. Tong, R. Laramee, G. A. Smith, P. Brookes, and T. D'Cruze, "Interactive analytical treemaps for visualisation of call centre data," in *Smart Tools and Apps for Graphics-Eurographics Italian Chapter Conference, G. Pintore and F. Stanco, Eds. The Eurographics Association*, 2016.

[36] R. Roberts, R. S. Laramee, G. A. Smith, P. Brookes, and T. D'Cruze, "Smart brushing for parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.

[37] R. C. Roberts, D. Rees, R. S. Laramee, P. Brookes, and G. A. Smith, "RiverState: A Visual Metaphor Representing Millions of Time-Oriented State Transitions," in *Computer Graphics and Visual Computing (CGVC)*, G. K. L. Tam and F. Vidal, Eds. The Eurographics Association, 2018, pp. 81–89.

[38] D. Rees, R. C. Roberts, R. S. Laramee, P. Brookes, T. D'Cruze, and G. A. Smith, "GPU-Assisted Scatterplots for Millions of Call Events," in *Computer Graphics and Visual Computing (CGVC)*, G. K. L. Tam and F. Vidal, Eds. The Eurographics Association, 2018, pp. 71–79.

[39] ——, "Feature-rich, GPU-assisted scatterplots for millions of call events," *Computers*, vol. 8, no. 1, pp. 1–21, 2019.
[40] T. Hogan, U. Hinrichs, and E. Hornecker, "The elicitation interview technique: Capturing people's experiences of data representations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2579–2593, Dec 2016.
[41] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *The Craft of Information Visualization*, ser. Interactive Technologies, B. B. Bederson and B. Shneiderman, Eds. San Francisco: Morgan Kaufmann, 2003, pp. 364 – 371.
[42] G. Ellis and A. Dix, "A taxonomy of clutter reduction for information visualisation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1216–1223, Nov 2007.
[43] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 3, pp. 1249–1268, 2016.
[44] D. A. Keim and H. . Kriegel, "Visualization techniques for mining large databases: a comparison," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 923–938, Dec 1996.
[45] M. C. Ferreira de Oliveira and H. Levkowitz, "From visual data exploration to visual data mining: a survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 378–394, July 2003.
[46] E. Dimara, A. Bezerianos, and P. Dragicevic, "Conceptual and methodological issues in evaluating multidimensional visualizations for decision support," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 749–759, Jan 2018.
[47] M. Williams and T. Munzner, "Steerable, progressive multidimensional scaling," in *IEEE Symposium on information Visualization*. IEEE, 2004, pp. 57–64.
[48] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in *Proceedings of the First IEEE Conference on Visualization: Visualization90*. IEEE, 1990, pp. 361–378.
[49] P. Embrechts and A. M. Herzberg, "Variations of andrews' plots," *International Statistical Review/Revue Internationale de Statistique*, pp. 175–194, 1991.
[50] J. Heinrich, Y. Luo, A. E. Kirkpatrick, H. Zhang, and D. Weiskopf, "Evaluation of a bundling technique for parallel coordinates," *arXiv preprint arXiv:1109.6073*, 2011.
[51] K. T. McDonnell and K. Mueller, "Illustrative parallel coordinates," in *Computer Graphics Forum*, vol. 27, no. 3. Wiley Online Library, 2008, pp. 1031–1038.
[52] H. Siirtola, "Direct manipulation of parallel coordinates," in *2000 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics*. IEEE, 2000, pp. 373–378.
[53] L. Wilkinson, A. Anand, and R. Grossman, "Graph-theoretic scagnostics," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* IEEE, 2005, pp. 157–164.
[54] D. Rees, "Vimeo AgentVis Demonstration," https://vimeo.com/325757055, uRL Accessed: 2019-03-21.
[55] ——, "YouTube AgentVis Demonstration," https://www.youtube.com/watch?v=hsD21tHybts, uRL Accessed: 2019-03-21.
[56] S. Lee, S. Kim, and B. C. Kwon, "Vlat: Development of a visualization literacy assessment test," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 551–560, Jan 2017.
[57] C. C. Gramazio, D. H. Laidlaw, and K. B. Schloss, "Colorgorical: creating discriminable and preferable color palettes for information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 521–530, 2017.
[58] Tableau Software, "Tableau," https://www.tableau.com/, uRL Accessed: 2019-02-27.
[59] A. C. Telea, *Data visualization: principles and practice*. Boca Raton, FL: CRC Press, 2014.
[60] C. A. Brewer, "ColorBrewer 2.0," https://www.ColorBrewer2.org, uRL Accessed: 2019-02-27.
[61] K. Moreland, "Why we use bad color maps and what you can do about it," *Electronic Imaging*, vol. 2016, no. 16, pp. 1–6, 2016.
[62] M. Stone, "How we designed the new color palettes in Tableau 10," https://www.tableau.com/about/blog/2016/7/colors-upgrade-tableau-10-56782, 2016, uRL Accessed: 2019-02-27.
[63] Sandia National Laboratories, "Paraview," https://www.paraview.org/, uRL Accessed: 2019-02-27.
[64] T. Munzner, "A nested model for visualization design and validation," *IEEE transactions on visualization and computer graphics*, vol. 15, no. 6, pp. 921–928, 2009.
[65] S. McConnell, *Rapid development: taming wild software schedules*. Redmond, WA: Microsoft Press, 1996.

Dylan Rees Dylan received his masters degree in automotive engineering from Loughborough University in 2006. He was awarded a masters in computer science from Swansea University specialising in data visualisation in 2016. Currently he is a PhD candidate at the Swansea University Department of Computer Science, specializing in data visualization.

Robert S. Laramee Robert received a bachelors degree in physics, cum laude, from the University of Massachusetts, Amherst (Zoo Mass) and a masters degree in computer science from the University of New Hampshire, Durham. He was awarded a PhD from the Vienna University of Technology, Austria at the Institute of Computer Graphics and Algorithms in 2005 (Gruess Gott TUWien). From 2001 to 2006 he was a researcher at the VRVis Research Center (www.vrvis.at) and a software engineer at AVL (www.avl.com) in the department of Advanced Simulation Technologies. Currently he is an Associate Professor at the Swansea University (Prifysgol Cymru Abertawe), Wales in the Department of Computer Science. His research interests are in the areas of scientific visualization, information visualization, and visual analytics.

Paul Brookes Paul is currently a lead developer at QPC Limited. Since joining Callscan Limited in 1989 he has continued to work on development of numerous computer telephony integration projects for contact centres providing real time and historic management information.His current interests include new and unique ways visualising of customer journey data for contact centres.

Tony D'Cruze Tony is a contact centre expert, with more than 20 years spent in a variety of contact centre roles in the Financial and Outsourcing sectors. Passionate about efficiency and driving improvements in customer service, Tony is now a Consultant with QPC and supports clients with leveraging QPC solutions to transform their customer contact operations.

Gary A. Smith Gary received his MBA from Leicester University in 2008 and a BSc in Mathematics & Statistical Computing from Liverpool John Moores University in 2006. He is also an Associate of the Operational Research society. Gary is currently the Director of Product & Marketing at QPC Group and is responsible for the strategic direction & innovation of QPCs products.

Aslam Miah Aslam has worked as a call-center agent for over two years with experience across sales and customer service departments.