RC24551 (W0805-051) May 9, 2008
Computer Science

IBM Research Report

Organizational Maps and Mashups

Danny Soroker, Aaron Zinman*, Chandra Narayanaswami
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

*MIT Media Lab
20 Ames Street
Cambridge, MA 02142

Organizational Maps and Mashups

Danny Soroker
IBM T.J Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
soroker@us.ibm.com

Aaron Zinman[*]
MIT Media Lab
20 Ames Street
Cambridge, MA 02142
azinman@media.mit.edu

Chandra Narayanaswami
IBM T.J Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
chandras@us.ibm.com

ABSTRACT

In this paper we introduce *OrgMaps*, an interaction design for navigating human organizations that is tightly coupled with people-centric data. Enterprises are increasingly in need to visualize the large number of its globally-connected members, but cannot without a common visual and analytical basis. OrgMaps attempts to become this basis, providing a flexible layered navigational structure and API on which to build new applications, similar in spirit to Google Maps and their API that have revitalized the medium of geography. Our strategy layers photos and data about members on top of the hierarchical icicle plot of the organizational structure in an intelligent zooming interface. Example data sources include people-centric attributes, such as list of publications, instant messaging status, performance ratings, and tags. We present the design principles and features behind OrgMaps, and argue that it and similarly integrated organizational tools will radically change the way organizations view themselves and use such knowledge.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors.

Keywords: Visualization, hierarchy, mashups.

INTRODUCTION

The efficiency of a globally integrated enterprise is increasingly dependent on how easily employees can find people-centric information to manage or measure processes, collaborate with others in the course of their work, follow interesting events, and the like. A wealth of people-centric information in the enterprise currently resides in structured databases or in unstructured repositories with few effective ways to dynamically associate and visualize the data in a fluid and interactive manner. Until a few years ago, location-based information was in a similar state, even though several desktop map software packages were available. The advent of highly interactive web-based maps using Web 2.0 techniques with published APIs for associating data through mashups, such as Google maps, changed the landscape of location-based information. Overnight, it seemed the sluice gates were opened, and a flood of location-based mashups became pervasive. Interactive crime statistics, rental properties, nearby inexpensive gas stations and street-level photos are newly accessible to everyday people by layering the external data on a common geographic meme and interaction design. We believe this has become possible due to a) a highly interactive visual map canvas/substrate where zooming and panning appeared instantaneous, b) simple and well-documented APIs that allowed data to be overlaid on maps, and c) devices such as GPS enhanced cellphones and cameras, and tools such as interactive maps that allowed people to tag information and events with location.

We sense a similar transformation beginning to occur in the enterprise. One potential substrate, the subject of this paper, shows the hierarchical organizational structure. A wide variety of people-centric information such as competencies, accounts or clients they recently worked with, publications, sales figures, blogs etc., can be overlaid and visualized on such a substrate. The ability to visualize such information as well as exploring the hierarchy itself is the focus here.

Just as with geographic maps and location-based mashups, organizational maps – *OrgMaps* – and people-centric mashups need a visual substrate that is fun to use and can be rapidly manipulated to hone in on relevant parts of an organization, in addition to mechanisms for associating people-centric information on the organizational map and tools to create people-centric information streams. When combined with search functions and compounded filters, OrgMaps allows employees and consultants to rapidly solve problems they face everyday.

Potential people-centric enterprise mashups on the OrgMap encompass several areas including HR metrics and processes, financial performance and budget tracking, collaboration and expert finding, CIO office and internal asset management, project staffing, enterprise workflow, and people-centric RSS feeds and tagging. The data itself can be associated with individual employees or be aggregated up the organization to produce rolled up status. Such mashups can be used to quickly visualize the state of an organization or a given subset. For example, managers who have not completed reviews with their employees can be detected in a single glance. Employees who have too many computing devices assigned to them or have not signed up for health benefits program can be similarly spotted.

We envision many usage scenarios for OrgMaps along a collaboration continuum. The simplest is a single user scenario where a user may view and navigate part of an or-

[*] Work done while at IBM Research.

ganization when creating a mashup for temporary use. On the opposite extreme, users create OrgMap mashups and share with who in turn continue to refine and share the mashup by applying filters or editing data. In between, several users may simultaneously view an existing OrgMap to collectively annotate and append data.

Realizing our vision requires solving challenges both technical and conceptual. The first requirement is to provide an appropriate visual and interactive substrate that is meaningful in a wide range of enterprise scenarios. It must remain intuitive and provide rapid navigation even for very large organizations and data streams. Scaling is both a technical problem of minimizing start up time and application code on the user's computing device, and a challenge in visualization strategy. Next, challenges for adoption of enterprise mashups lie in privacy concerns and organizing data feeds. For example, HR related applications are particularly sensitive to privacy, and thus the data formats will need some inclusion of access controls and protocols. Further, getting organizations to structure their data as people-centric feeds is yet another challenge. We believe that this transition will happen as enterprise visualizations and network-based mashups become more prevalent. For example the technical reports published by IBM Researchers are now available as an RSS feed. As more Web 2.0-like tools become adopted we expect that employees will be motivated to produce or convert their data to RSS-like mechanisms. Developing a standardized format for RSS feeds with self-described privacy and access controls with enforcement mechanisms can help spur wider adoption.

With the larger problems and direction in scope, we now present our current proposed system, which we have prototyped, for organizational mashups. We first lay out the system's more immediate requirements. We then present the chosen visualization and interaction design. This is followed by a discussion of the types of mashups that can be visualized at present, how data for mashups enters the system, and further address issues surrounding interoperability and deployment. We conclude with a review of prior work in this area, feedback from our demonstrations, and ongoing work.

SYSTEM DESCRIPTION

Requirements and Goals

In order to construct a functional interactive organizational map, we needed to establish guidelines that fit our intended user goals:

1. The ability to gain global impressions while exploring local details of an organizational hierarchy, transitioning between the two ends smoothly and rapidly.
2. The ability to easily associate, overlay, and visualize various forms of data contextually in the organizational hierarchy.

Our requirements led us to the following concrete interaction design constraints:

1. Simple layout of the hierarchy.
2. Fluid zoom/pan interaction.
3. Generic mechanisms for associating data.
4. Simple metaphors for visualizing overlaid data.

Our chosen visual design builds upon one of the simplest hierarchical layouts, the icicle plot [12]. Icicle plots place parents directly above their children, keeping edges implicit rather than explicit. In this way, the plot can be called space-filling. Each node is represented by a rectangle whose width is the sum of the widths of its children. All nodes have the same height, and all leaf nodes have the same width (for a given zoom level). An example diagram is shown in Figure 1.

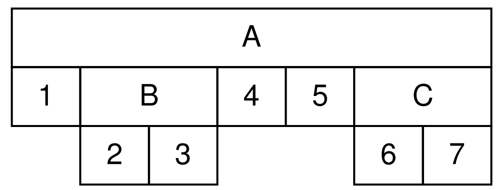

Figure 1: Sample icicle plot

This figure shows a small organization with 10 people, consisting of 3 managers (nodes A-C) and 7 non-managers (nodes 1-7). The reporting structure is very easy to grasp by glimpsing at the figure (e.g., 3 reports to B, and B reports to A). The ability to follow parentage vertically is a primary reason to choose icicle plots over alternative layouts. Advantages of the icicle plot over alternative layouts for the task at hand are elaborated upon in the Discussion section.

In the following sections, we present the different visualizations supported, detail the interaction mechanisms, and discuss the mechanisms and data formats used for generation of people-centric mashups.

Visualization Overview

Base Visualization

Figure 2 shows a screen shot of our interactive implementation of an icicle plot for organizations, OrgMaps. The screen shot demonstrates a fictitious organization of 150 people, renovations.com[2].

OrgMaps uses faces as a central aspect of its visualization. It builds upon human ability to quickly recognize faces and thus help form a visual memory of the organizational structure that a user builds up over time. As the entire organization is visible, leaf nodes become very thin. Only nodes that are wide enough (beyond a threshold we set) show the face of the person they represent (in this case, 14 of the 150 faces are visible). However, by instrumenting OrgMaps with zoom and pan mechanisms, we can investigate all branches and individuals in a method similar to starting with a map of the US, zooming in to a city, and then panning to locate its various neighborhoods. Via a user interface gesture we can zoom in on a person so that they be-

[2] See [10] for an interesting discussion of "the rule of 150".

come the focus of the plot, as shown in Figure 3. Note that, when zoomed in, the complete management chain is still shown, so that context is maintained and can be used for navigation.

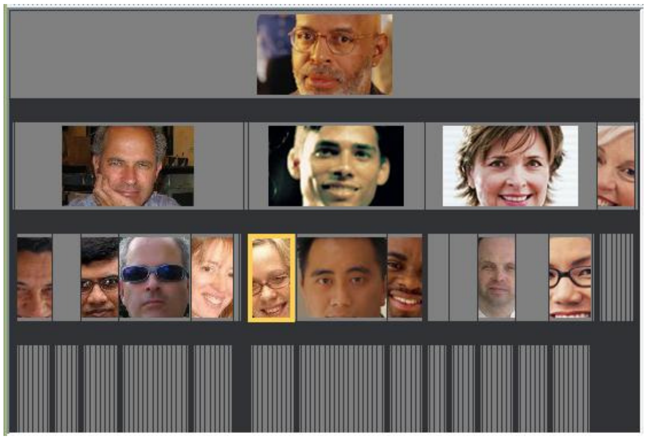

Figure 2: Global view of an organization

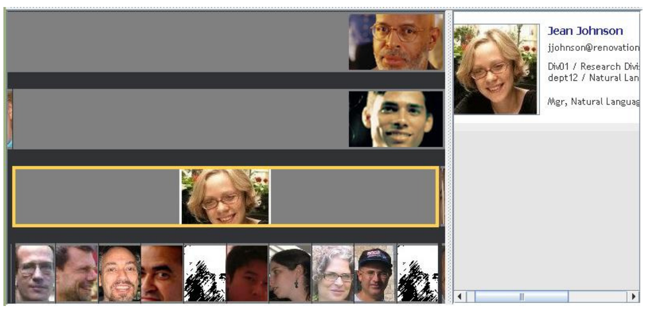

Figure 3: Zoomed-in view of a department

This figure also shows the details panel to the right of the plot, in which information about the selected person is presented. Both the faces and displayed information are obtained from a centralized corporate directory.

Visualizing Mashups

In its current form, OrgMaps supports Boolean and scalar variables for visualizing node attributes. The actual data source may show more data in the details view per person. Figure 4 depicts an example Boolean mashup, showing, for each person, whether or not they have enrolled for benefits this year (such as health insurance).

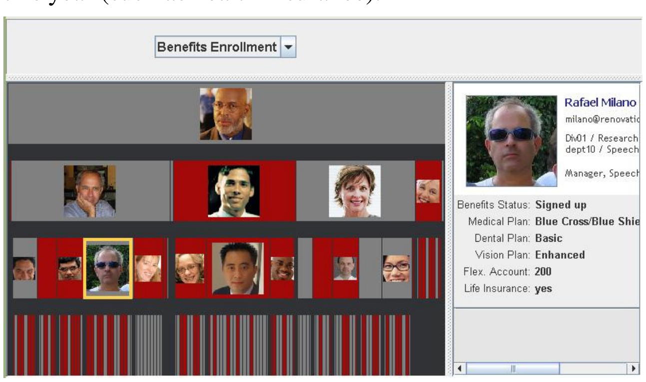

Figure 4: Boolean mashup; faces are shrunk

Because the mashup was customized to highlight "trouble" regions, red cells highlight individuals who have not signed up for benefits. Such people are easily found through gestalt. Further details regarding benefits of people who have signed up can be found in the details panel. Note that, in contrast to the previous screen shots, faces were shrunk so that the mashup values would be easier to see.

Figure 5 shows a scalar mashup of the organization's patents.

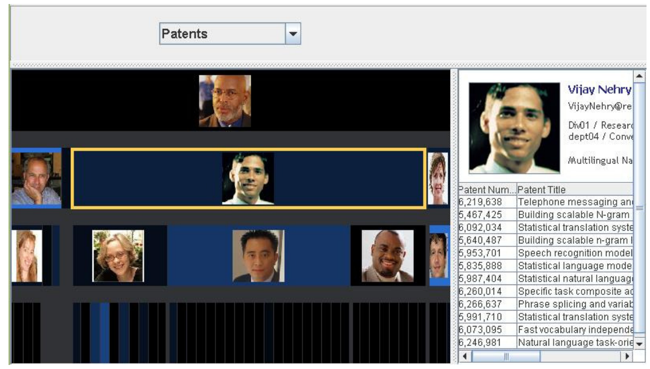

Figure 5: Scalar mashup

This mashup uses color intensity to reflect the number of patents, where a brighter blue background reflects more patents. All patents of the selected individual are displayed in the details view. We obtained real patent data for IBM employees from the USPTO web site.

Visualizing Connections

Beyond organizational structure, we often wish to see other connections within an enterprise. As an example, OrgMaps lets us visualize patent co-inventors on the map by highlighting the selected person in one color (yellow), and the co-inventors in another (pink), as shown in Figure 6. The list of co-inventors is also displayed in the results dialog (on the bottom right). Scrolling through the list modifies selection on the map and the details panel accordingly.

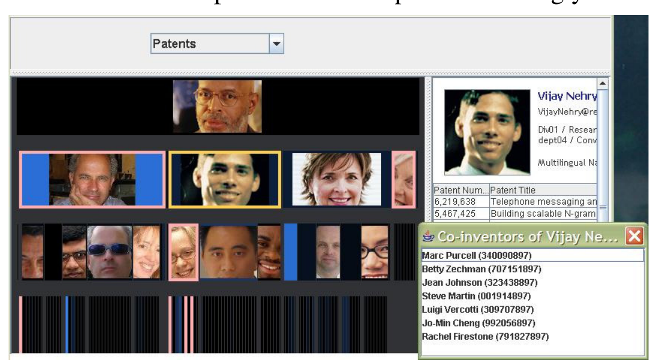

Figure 6: Interpersonal connections

OrgMaplets and Overview View

The screen shots thus far show detailed views of an organization in its entirety. Our system enables instantiation and viewing of multiple organizational snippets, called OrgMaplets. Figure 7 features three OrgMaplets, where the bottom two were created from the larger map on top.

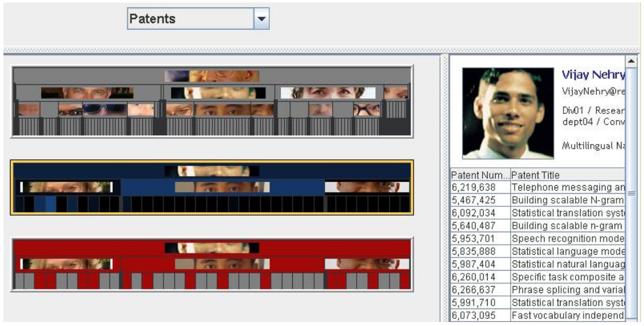

Figure 7: Overview view

In this example, the OrgMaplets are created to expand select portions the organization. Aside from structural zoom, each OrgMaplet reflects a different mashup as specified by the user, enabling multiple views into the organization.

Interactions

OrgMaps currently supports the following interactions:

Selection: the selected person changes dynamically – when moving the mouse, the hovered person becomes selected – they are highlighted on the map and their data is shown in the details panel. To freeze the current selection, a single mouse click "locks" the map, so that mouse movement does not affect selection. An additional click or escape unlocks. Due to this highly responsive behavior, our interactions try to avoid moving the mouse across the map (as would be needed by a window menu or scroll bar) and instead are "local" interactions – context menu, keyboard actions, and mouse clicks and drags. For example, pressing the shift key while moving the mouse rapidly pans through the entire organization, even at high zoom levels.

Zoom, pan, traversal: dragging the map horizontally pans it, whereas a vertical drag zooms in and out. The system detects whether the drag is horizontal or vertical, modifies the cursor accordingly, and proceeds with the operation. Plus/minus keys are used for zooming, and the arrow keys are used for traversing the tree. An "optimal zoom" feature scales the selected node to occupy the entire available width (as shown for Jean Johnson in Figure 3), or just wide enough so that it and all its descendents show faces.

Search: people can be searched for either globally or contextually (i.e., within the subtree rooted at the selected node) by various fields such as name or e-mail. Data from mashup sources can also be searched (such as searching for all patents whose title contains "speech" within a certain department).

View controls: a menu entry lets you specify how faces are rendered: normal, shrunk, or hidden (hiding faces is effective for revealing mashup values, such as when browsing an organization for people with lots of patents, and also may be used as an anonymization mechanism for addressing privacy concerns). Another menu entry controls whether the details panel shows only data of the currently-selected mashup or aggregation of the data from all loaded mashups.

OrgMaplets: a menu entry creates an OrgMaplet rooted at the selected person. The overview view contains menu entries for creating an OrgMaplet from a file cached locally, or fetching the organization rooted at a specified individual dynamically from an LDAP server.

Information and linkage: some menu entries provide additional information about the selected individual, such as the size of their organization. Others provide linkage, such as opening a browser with the selected person's web page (or enterprise directory entry) or sending e-mail to the selected person or to their entire department.

Data Feeds and Mashup Generation

Mashups are populated through data feeds that contain information on people in the organization. Similar to geographical mashups for which data is associated via its location, here data is associated with a person via their unique identifier, (e.g., employee serial number or e-mail address). In our research prototype, and for easy interoperability, we support CSV formatted files (comma-separated values) for data feeds. This is a simple lowest-common-denominator ASCII format, compatible with spreadsheets such as Excel. Future enterprise deployments likely will alternatively use organizational databases and other existing data sources. A mashup handler reads in a mashup feed, associates data elements with corresponding nodes of the organization, and registers the mashup with the OrgMap. All registered mashups populate the drop-down menu at the top of the OrgMaps applet area, and, upon selection by the user, are visualized on the OrgMap. We are designing a self-describing format for feeds, in which the header specifies its type and association so that the selection of mashup handlers can be better automated.

Each mashup handler has an associated node painter and details renderer. The node painter is added to the OrgMaps painting pipeline when the corresponding mashup is selected. In the examples shown above, mashup node painters affect the background color of rendered nodes (the Boolean mashup renderer uses two colors to represent true vs. false; the scalar mashup renderer uses color intensity to represent numeric value). The details renderer contributes mashup-specific information to the details panel (such as a list of patents for the selected person).

We also support the CSV format for "OrgFeeds" – a serialized form of the organizational data itself containing the details of each person and who they report to.

Scenarios

We envision that OrgMaps can be operated upon by several people either simultaneously or used sequentially in a cascaded fashion. Consider the following example. An attorney in the patent licensing department is looking for experts to evaluate a group of patents that have just been issued. The attorney may start by searching for people in the organization who have issued patents in the same area, and then forward the resulting visualization and data objects to the managers of the employees with the right expertise to solicit feedback. Managers may receive only sections of the Org-

Map for portions of organization under their purview, or people they have supervised in the past.

Once the responses come in, the attorney may use the map to directly initiate contact with the employees. Such contact can include direct access to email, instant messaging or telephone contact with the selected employees. In this example, different people in the enterprise collaborate and apply their area of expertise to solve a problem together. We used the manager's day to day knowledge of his employees, which is unlikely to be in an enterprise database, to avoid automatic selection of employees by the attorney and subsequent generation of unwanted e-mail. The system can also remember employees who have cooperated in this patent evaluation process in the past and in future either target them since they are responsive or avoid them to spread the work around. Such information may be shown on the OrgMap to the attorney and to the manager before allocating these tasks.

In another scenario, an OrgMap can be used by the sender of a broadcast calendar invitation to see who accepted the invitation and look for any patterns in the responses. Are members of the molecular storage department interested in a talk by anthropologists?

OrgMaps may also be used to detect patterns in evolving data. For example, an HR employee may view weekly snapshots of the benefits enrollment feed to get a better sense of the dynamics of the enrollment process. This can be combined with e-mail reminders to a selected list of managers directly from the OrgMap.

Interoperability and Deployment

OrgMaps is intended to be part of an arsenal of organizational productivity tools. With an eye towards broader deployment, we have been pursuing the following directions:

Connection to other tools

Since OrgMaps should interoperate with other tools, and we don't want to replicate their existing functionality, OrgMaps should provide seamless connectivity to myriad tools. We have already provided linkage to the IBM corporate directory and e-mail. We are pursuing connectivity to the corporate instant messaging system – for viewing the status of people within an OrgMap, for viewing current chat partners and for launching chat sessions.

Incorporation into a mashup authoring system

Lotus Mashups [14] is a lightweight mashup environment for rapidly assembling Internet and enterprise content into simple, flexible, and dynamic Web applications. We incorporated a version of the OrgMaps applet as a widget for Lotus Mashups. This widget accepts various data feeds as input, both for the organizational structure and for mashup data, and sends selection events as output (name or e-mail).

Consider the following scenario. An OrgMaps widget is placed on the Lotus Mashups canvas, and is initially empty. An org feed is dragged onto it, at which point the widget displays the organization. Next, a data feed (e.g., benefits enrollment) is dragged onto the widget, whereby the mashup is displayed on the map and added to the drop-down list of mashups. Now another widget is placed on the canvas, say one that shows extended profile information. We connect the OrgMaps widget to the profile widget using the "send e-mail" event action. Now, when we select a person in the OrgMap, the profile widget gets updated to show their profile. This scenario is captured in Figure 8, and shows the power of OrgMaps to seamlessly connect with external data sources to quickly create mashups.

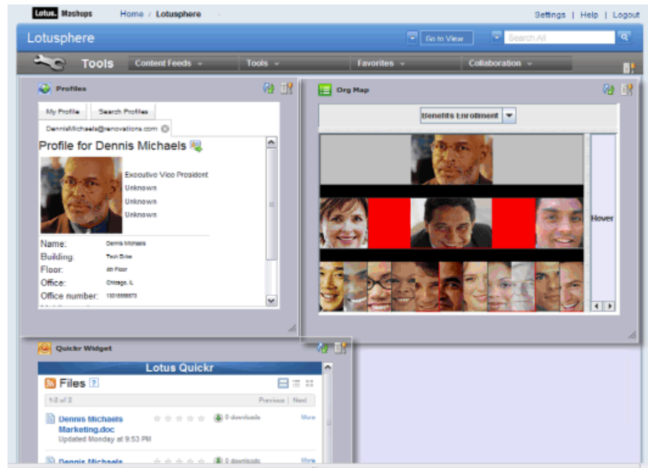

Figure 8: OrgMaps widget in Lotus Mashups

Browser-based implementation

A browser-based AJAX implementation of OrgMaps is in progress. We expect that a native browser-based implementation would be more consumable for widespread deployment. Architecturally, developing this version leads to a cleaner separation between client and server. As part of this work, a server that dynamically fetches parts of the organization per user specification has been developed.

DISCUSSION

Design considerations

Why icicle over other tree layouts?

After careful review of existing related work of hierarchical visualizations and techniques, we developed a set of criteria we hypothesize necessary for interactive visualizations of organizations. First, we wanted the visualization to be extremely legible and intuitive, as this was intended to be quickly picked up by users in the same lightweight manner one may use Google Maps. Therefore, the learning curve had to be as close to zero as possible (eliminating Treemaps). Second, because we expect many of the queries to search for "hot spots" over large regions of the tree (tens of thousands of people), we needed the entire global view to be present (eliminating hyperbolic layouts), unoccluded (eliminating 3D cone layouts ([3], [18]), and revealing as much of the structure of sub-regions as possible (eliminating degree-of-interest trees). Third, to maximize real estate for potentially large trees, we seek to eliminate unnecessary structural elements such as edges, instead preferring the structure to be implicit in the visual scheme and hence space-filling. Fourth, we wanted the ability to easily navigate across the same level of the organization, even while zoomed in (eliminating radial layouts). Last, we needed the attributes and photos of each node shown alongside the nodes in the hierarchy. The icicle plot meets all of the above criteria when used in conjunction with pan-and-zoom mechanisms.

Beyond use of color for mashups

Icicle plots themselves are limiting. Schemes such as segmenting the boxes horizontally to show additional attributes per-person can only scale so far. After enough slicing, the plots will loose legibility. One alternative is to change the shape to better represent the individual, such as Gist Icons [4]. Another would be the ability to transition the icicle plots to alternative visualizations, as suggested by Fry [7]. However, until users develop fluency with the current schema, we felt simplicity was more important than presentation flexibility.

Another thought is to use nodes that are further up the hierarchy with larger display width to show aggregate information for the sub-organization they represent. For example, the node area may be used to show a histogram depicting the set of individual values in a scalar mashup. It may make sense to turn this feature on selectively for a subset of the nodes, so as to avoid visual clutter.

Face rendering

As OrgMaps visualizes people and their associated data, we prioritize the display of portraits to aid in navigation. However, many boxes have little room to draw the face. We exploit the user's cognitive machinery for human faces by preferring, as default, to show detail over area of the face. Cropping in such a way is more effective for face recognition even though only the nose and mouth might be shown. The user can optionally choose to view shrunken complete portraits, as shown in Figure 4, or hide faces altogether.

Faces are rendered in the main view only when their corresponding node is visible and wide enough (the current threshold is 30 pixels), as shown in Figure 2. This design choice eliminates extraneous visual clutter (by keeping narrow spaces clean rather than filling them with an unrecognizable face fragment) and is crucial for performance when rendering thousands of nodes. Hiding faces for narrow nodes also helps perceive mashup information.

Faces are, by default, drawn in the center of a node. However, if the center area of a visible node is hidden, i.e., outside the viewing window, for a given zoom/pan configuration, its face is shifted to a visible location (e.g., the top two nodes in Figure 3). If the visible area of a partially-visible node is narrower than the width of the face image, the face is "squished" along the X axis (as can be seen for some nodes in Figure 5).

Faces are obtained dynamically from a corporate directory or other repository, and are cached locally for improved performance and offline use. If a face is unavailable, we use a generic substitution. Each face image is loaded asynchronously when it is first visible and thus needed. As faces appear for the first time in the main view, they make a faint popping sound, to provide audio feedback and amuse the user.

Experience

We have demonstrated the evolving OrgMaps prototype internally at IBM and also at Lotusphere earlier this year. The IBM corporate directory, which provides profile and organizational information, is one of the most heavily used internal applications. Consequently, there is clear interest in technology such as OrgMaps, which provides complimentary views and the ability to rapidly access aggregate data. Some people from other organizations said that the management structure plays little role in their domain, and that functional and social ties are more relevant. However, in most cases and even for relatively small organizations, participants clearly agreed that the organizational structure is a meaningful view.

One answer to those less interested in the formal organizational structure is the following: OrgMaps specializes in viewing hierarchies, but the hierarchy need not be based on the employee-manager relationship. For example, a map for an elementary school may have nodes for students, grouped by teachers, grouped by grade levels (which do not correspond to a person), and so on. In a corporate organization, the hierarchical map may be location-based.

Evaluating the interaction mechanisms requires "test driving" by users unfamiliar with OrgMaps. In our experience to date, it appears that the highly dynamic behavior is a two-edge sword, possibly due to the overly simple single-click locking mechanism. The problem is that the system enters locked mode without the user intending it to do so. Once users become familiar with this behavior, it ceases to be problematic. To aid new users we plan to add visual cues when entering and leaving locked mode.

We have already incorporated several changes from feedback at our demonstrations. One suggestion was to keep the faces visible for all visible nodes that are large enough, which led to the concept of floating and squished faces, as seen in Figures 3 and 5. Another comment noted that faces that take the entire height of a node seem to visually break a node into three separate nodes (left, face, right). In response, faces are now fully embedded in nodes (with margins on top and bottom). Another suggestion was to use heated object spectrum for scalar mashups, as an improvement for color-blind users. Upon experimentation we decided not to use this as the default rendering, since different hues in combination with the highly segmented nodes create too much visual complexity. That said, this raised our awareness of the importance of adding user controls for tailoring the view.

CHALLENGES AND ONGOING WORK

The main focus of our current work is to evolve OrgMaps from a research prototype to a deployed system. Here are some of the challenges we are addressing for a successful transition.

Scalability and performance

Our prototype handles organizations with thousands of people at interactive speeds (load, zoom, pan, and search). We have tested the system on organizations within IBM of up to 35,000 people. Panning an entire organization with tens of thousands of visible nodes is not enjoyably fast (though once zoomed-in, interaction is snappy). The main challenge is how to render approximations of icicles with very many nodes without losing important details (such as the depth of the tree in various parts of the organization). Our code contains optimizations for collapsing blocks of tiny adjacent nodes, by understanding that we should bubble notable attributes to the aggregate. Additional sophistication (such as a method for estimating where to draw blocks and gaps) is needed for handling significantly larger hierarchies.

Another performance bottleneck may arise when requesting a large organization to be fetched from a centralized server. One approach we are exploring, aside from caching, is successive refinement, bringing in gradually more and more layers of the organization.

Mashup data feeds

In the current prototype, existing feeds are associated with particular mashup handlers (e.g., benefits enrollment with a Boolean handler, and patents with a scalar handler). Looking forward, as we add new types of mashup handlers the need arises for feeds to become self-describing to help the handler determine how to best use the data. This should be aided by the user through previewing and interacting with the data sources.

Another important enhancement is handling of live data feeds, such as presence. Such data can be read on demand, when a person is selected or their node becomes visible, similarly to the way faces are loaded in the current system.

The validation and consistency of data feeds obtained from various sources is important for accurate visualization. Supporting validation of feed content is an important aspect of any mashup system. Besides the data itself, its correct association to people is challenging, since the externally known information on a person is often just their name, and names are not unique.

Other domains

OrgMaps specializes in rendering hierarchies, not necessarily organizational hierarchies. We have started to explore the application of its principles for viewing data that is organized into categories and sub-categories. Two ways in which this problem extends the domain of our current prototype are that nodes become non-uniform and the hierarchy of a given set of objects is no longer fixed, as it depends on the classification order. For example, we would need to provide interface gestures for "pivoting" between classifications.

RELATED WORK

Creating an effective visualization of large hierarchies, like most visualizations, is a matter of trade-offs ([1]). Because there are more data points than pixels on the screen, making the choice of trade-offs should be based on the user's goals. One principal dichotomy in visualization is the balance between conveying an overall impression of the data versus zeroing in on a particular local region or intended destination. The ideal visualization is able to move fluidly between the two [7], but most focus on just one side of the problem.

Global views are useful to gain impressions of the data's overall characteristics, often used to find "hot" regions of interest. Global views are most useful when the user is not seeking a node in the tree that is known *a priori*, but rather starts from the maximum global viewpoint and follows an information scent [17] afforded by visualization strategy to home in on a local destination. Often, one cares more about either the structure of a hierarchy (or graph) or the attributes of the nodes contained in it. This choice is important, because space-filling visualizations such as Treemaps [11], icicle plots [12], and radial layouts [19] do away with extraneous structural details such as lines connecting nodes, which are desirable when explicitly looking at structure. By contrast, space-filling visualizations are designed to maximize attribute gestalt by making the linked structure implicit.

Standard graph-drawing routines for directed graphs, such as circular layouts [8] or energy-minimizing layouts [5], often do not scale well to large sizes. Some techniques that prioritize communication of the overall structure within the available resolution support scaling by reducing the graph topologically [9]. While such techniques are useful for telecommunication backbone visualization, they are less useful within organizational hierarchies because their shapes look predictably uniform. Furthermore, such layouts are difficult to augment with node attributes (like size, color, shape) in comparison with Treemaps or icicle plots, because so much space is wasted on visual elements outside of the attributes.

Treemaps use a divide-and-conquer methodology to compact a dendrogram into a series of rectangles embedded within each other in smaller and smaller regions. The region for each node contains the regions of all of its descendents. Treemaps principally afford mapping of data onto the hierarchy through the color and size of each rectangle. Color provides a general gestalt effect which Treemaps use to highlight regions of interest, of which the effect is non-linearly (in perception) modulated by the size of the rectangle. Even though they are compact and provide a strong global perspective, Treemaps suffer from legibility issues [11] due to the non-obvious distortions in the hierarchies. As elaborated by Lü and Fogarty [15], an important limitation of treemaps is the difficulty of discerning the structure of a hierarchy. Further, by having simple regions for leaf nodes but not for internal nodes, Treemaps are not suitable for people-based hierarchies in which we want to easily reveal attributes of all nodes.

Radial layouts are also space-filling, and use concentric rings to represent each level of a hierarchy. They visually operate with the same principle as icicle plots, but increase the relative size per layer as depth progresses as a function of being radial. This is especially useful in person-centric hierarchies because the number of people per level in the hierarchy typically increases, giving more real estate where it is most needed. In contrast to icicle plots, it is difficult to trace along the members of a single layer in a radial layout, a problem which is most exaggerated when zoomed in. Similarly, it is harder to sense visual orientation when traversing up the hierarchy. Like Treemaps, radial layouts provide strong gestalt for node attributes using color and size, and are amenable to interactive focus+context techniques [18].

Any chosen layout not only affects the type of gestalt within a global view, but also provides differing affordances for following nodes along their structural paths. The most common method of enhancing the user's understanding of where to next proceed is focus+context, where focal points are shown adjacent to their larger immediate context.

The pan-and-zoom mechanism we have employed in OrgMaps is a basic use of focus+context. It is effective through its simplicity in implementation and concept; the learning curve is very low while maintaining high functionality.

Another popular focus+context theme is fisheye distortions. Fisheye techniques apply hyperbolic, or "fisheye lens", distortions to fluidly move from focus to context, often yielding or in conjunction with a larger global view. Fisheyes can be awkward and problematic when global views become particularly large; relatively small magnification levels do not provide adequate focus, while in large magnification levels one can quickly become lost when a small hand movement translates into a radical visual translation.

Lamping and Rao's hyperbolic layout achieve an extreme version of the fisheye lens by projecting trees radially onto a hyperbolic surface [13]. Because the hyperbolic transform only shows a very small portion of the tree (usually not more than 2 hops), one is able to fly through local neighborhoods with speed, legibility, and ease. However, they cannot give a global impression.

Another popular focus+context strategy, implemented by degree-of-interest trees, attempts to alleviate the problems with fish-eye distortions by instead operating at the topographical level. By watching the user and guessing their intent as they navigate along a tree, the visible topology is altered to reduce the seemingly uninteresting areas and expand the desired. Such manipulation can be purely automatic [2], or manual [15]. DOI trees tend to work best when a strong information scent is present [17]. Because they hide large portions of the tree, they are ineffective at providing global impressions. Further, the graphical representation of collapsed sub-trees often does not provide cues as to their sub-structures outside of relative size [16], or distributions of node attributes.

Mashups

Mashups are a popular emerging genre of media for combining disparate data sources. One key characteristic for a successful mashup is the ease of data manipulation on top of the mashable substrate. Google Maps is one of the most popular mashable substrates, in part because the map implementation is well-executed and relatively novel in accessibility, but also because Google provides a substantial amount of flexibility and power in their APIs to join data and interaction on their maps. In addition to Google's programming APIs, Google Maps can use RSS and ATOM feeds as data sources. Thus any blog or "Web 2.0-compliant" media that can export location-based data through the various popular data formats can join the Maps interface.

Going beyond maps, the number of mashup editors is rapidly increasing. Mashup editors use visual programming techniques to join data sources and mashable substrates without requiring the users to know how to program in code, although frequently they combine the two. Popular mashup editors include Microsoft's Popfly, Yahoo Pipes, and IBM's QEDWiki and the upcoming Lotus Mashups [14] mentioned earlier in the paper. These editors can provide the data sources, filtering mechanisms, and the link to substrates (more RSS feeds, maps applications, basic visualizations, etc). While the interfaces are still fairly technical, in due time simpler and easier interfaces will make such programs accessible by everyday users. We see this as particularly important for enterprise applications, where users will want to take their spreadsheet data and combine it with other company resources to produce powerful situational applications that help address their challenges.

Participatory collaboration hinges upon easy-to-use tools that also make it easy to enter data. Many Eyes [20] enables powerful collaborative data analysis by letting users easily create and share visualizations directly from spreadsheets. By supporting in OrgMaps a similar interface for CSV data feeds, we hope to promote similar participatory uptake in the domain of enterprise mashups.

Another emerging trend, while not technically a mashup, is the copy-and-pasting of HTML code/URLs to easily embed outside data sources and web applications. Popular in MySpace and with YouTube, the ability for everyday people to take part of what would otherwise be a closed web application and move parts around follows the same logic as mashups. This is similar in rationale to Microsoft's OLE or Apple's OpenDoc from years ago, except simplified with web standards. We see such a notion important for future business applications, especially in the ability to copy and paste stateful URLs into IM and e-mail for collaboration.

Task proxies [6] provide a lightweight visualization of the completion state of tasks across an organization (such as our benefits enrollment scenario). Their emphasis is to easily see and share the evolving status data, but their visualization (as a honeycomb) is too terse and abstract for serving as a legible map of the organization. In lieu of zoom/pan interaction they support structural embossing in response to mouse-over events.

CONCLUSION

This paper introduces OrgMaps, a new paradigm for creating a common visualization and navigational strategy for understanding organizations. We hope that OrgMaps and related future work can achieve this goal in order to enable a standard for new applications to be built on, similar to how Google Maps provide a basis for understanding the geographic world. Once that vision is realized, members of an enterprise's community will be empowered to merge disparate data sources about their organization for self-reflection and day-to-day problem solving.

ACKNOWLEDGMENTS

We'd like to acknowledge Daniel Coffman, who is leading the work on building a JavaScript version of OrgMaps, and Chris Collins, Mikael Orn, David Taieb and Jon Harris for useful discussions during the course of this work.

REFERENCES

1. Barlow, T.; Neville, P., "A comparison of 2-D visualizations of hierarchies". Information Visualization, 2001. INFOVIS 2001, pp.131-138, 2001.
2. Budiu, R.; Pirolli, P. L.; Fleetwood, M.; Heiser, J., "Navigation in degree of interest trees". 8th International Working Conference on Advanced Visual Interfaces (AVI 2006); 2006 May 23-26; Venice; Italy. NY: ACM; 2006; 457-462.
3. Cockburn, A. and McKenzie, B. "An Evaluation of Cone Trees". In People and Computers XIV: British Computer Society Conference on Human Computer Interaction 2000, pp. 425-436. Springer-Verlag.
4. DeCamp, P., Frid-Jimenez, A., Guiness, J. and Roy, D. "Gist Icons: Seeing Meaning in Large Bodies of Literature". IEEE Information Visualization Conference, 2005.
5. Ellson, J., Gansner, E., Koutsofios, E., North, S., and Woodhull, G., "Graphviz and dynagraph – static and dynamic graph drawing tools". In Graph Drawing Software, M. Junger and P. Mutzel, Eds. Springer-Verlag, Berlin, 127–148. 2003.
6. Erickson, T., Huang, W., Danis, C. and Kellogg, W. A., "Social Proxy for Distributed Tasks: Design and Evaluation of a Working Prototype". *Proc. CHI'04 Human Factors in Computing Systems* (April 24–29, Vienna, Austria), ACM/SIGCHI, NY, 2004, pp. 559-566.
7. Fry, B. "Computational Information Design". PhD Thesis, Massachusetts Institute of Technology, 2004.
8. Gansner, E.R, Koren, Y. "Improved Circular Layouts". Lecture Notes in Computer Science, Graph Drawing, vol 4372/2007, pp. 386-398, Springer Berlin / Heidelberg, 2007.
9. Gansner, E., Koren, Y., North, S. "Topological Fisheye Views for Visualizing Large Graphs". In Proceedings of IEEE Visualization Conference, IEEE (2004), pp. 175-182, 2004.
10. Gladwell, M. *The Tipping Point*. Back Bay books, 2000.
11. Johnson, B., and Shneiderman, B., "Treemaps: A SpaceFilling Approach to the Visualization of Hierarchical Information Structures". *Proc. 2nd International IEEE Visualization Conference*, IEEE (1991), 284-291.
12. Kruskal, J.B, and Landwehr, J.M., "Icicle Plots: Better Displays for Hierarchical Clustering". The American Statistician, vol 37, no 2. pp. 162-168. American Statistical Association. 1983.
13. Lamping, L., and Rao, R., "The Hyperbolic Browser: A Focus+Context Technique for Visualizing Large Hierarchies". Journal of Visual Languages & Computing Volume 7, Issue 1, March 1996, Pages 33-55.
14. Lotus Mashups: http://www-306.ibm.com/software/lotus/products/mashups/
15. Lü, H. and Fogarty, J., "Cascaded Treemaps: Examining the Visibility and Stability of Structure in Treemaps". Graphics Interface 2008, to appear
16. Plaisant, C., Grosjean, J., and Bederson, B., "SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation". In Proceedings of the IEEE Symposium on information Visualization (infovis'02) (Oct. 28-29, 2002). INFOVIS. IEEE Computer Society, Washington, DC, 57.
17. Pirolli, P., Card, S. K., and Van Der Wege, M. M., "The effect of information scent on searching information: visualizations of large tree structures". In Proceedings of the Working Conference on Advanced Visual interfaces (Palermo, Italy). AVI '00. ACM, New York, NY, 161-172.
18. Robertson, G., Mackinlay, J.D., Card, S., "Cone Trees: Animated 3D Visualizations of Hierarchical Information". In Proceedings of the ACM CHI 91 Human Factors in Computing Systems Conference, pages 189-194, April 28 - June 5, 1991, New Orleans, Louisiana, June 1991. ACM.
19. Stasko, J.; Zhang, E., "Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations". In Proceedings of the IEEE Symposium on Information Visualization 2000, pp.57-65, 2000.
20. Viegas, F.B. Wattenberg, M. van Ham, F. Kriss, J. McKeon, M., "ManyEyes: a Site for Visualization at Internet Scale". IEEE Transactions on Visualization and Computer Graphics, Volume: 13 (6), pp. 1121-1128.