Technical Report no. 2005-12

TrustNeighborhoods: Visualizing Trust in Distributed File Sharing Systems

Niklas Elmqvist *Philippas Tsigas*

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computer Science & Engineering
Chalmers University of Technology and Göteborg University
412 96 Göteborg, Sweden

Göteborg, 2005

Abstract

We present TrustNeighborhoods, a security trust visualization aimed at novice and intermediate users of a distributed file sharing system. Building on the idea of "circles of relationships" as a model for computer usage proposed by Shneiderman, the TrustNeighborhoods technique uses the metaphor of a city to intuitively represent trust as a simple geographic relation, starting from the safe confines of one's own home to our own street, neighborhood, and the whole city itself. This metaphor gives users an easy way to relate to the trust of a document assigned by themselves as well as other users in the system. The visualization uses a radial space-filling layout; there is a 2D mode for editing and configuration, as well as a 3D mode for exploration and overview. In addition, the 3D mode supports a simple animated "fly to" command which is intended to show the user the context and trust of a particular document by zooming in on the document and its immediate neighborhood in the 3D city. The visualization is intended for integration into an existing desktop environment, connecting to the distributed file sharing mechanisms of the environment and non-obtrusively displaying a 3D orientation animation in the background for any file being accessed over the network. Informal testing indicates that the technique is a useful and intuitive way of visualizing trust in distributed file sharing networks.

Keywords: trust visualization, security visualization, circles of relationship

1 Introduction

Distributed file sharing systems are now commonplace in today's Internet-connected society and are a great way for users across the world to share and exchange information between each other. However, it is clear that security is a vital aspect for this kind of file sharing to succeed. As we continue to blur the border between local files and remote files on the network, it is becoming increasingly important to categorize files according to their *trust*, a function of the average trust of their owners as well as the owner's own classification of the file. To compound this problem, the majority of the intended users for the new generation of file sharing systems do not necessarily possess a high level of technical knowledge, and can be seen as novice or intermediate computer users. It is clear that we must find ways to make the concepts of trust and security explicit even to such relatively inexperienced users.

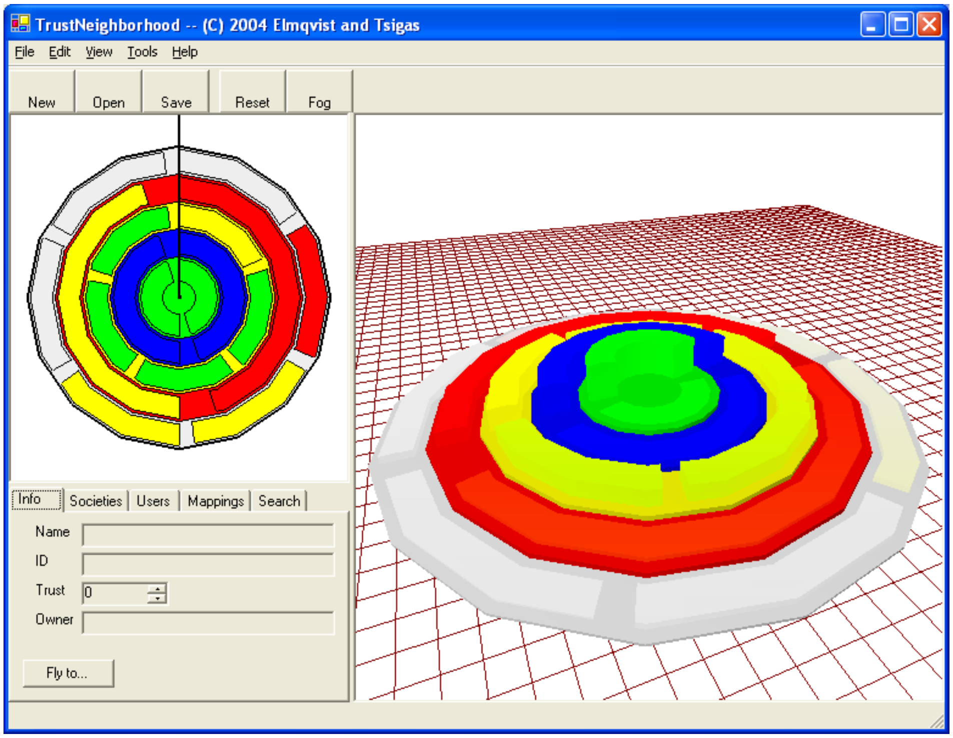


Figure 1: TrustNeighborhood visualization for a simple system with four users and 13 files organized into five societies.

In this paper we present TrustNeighborhoods (see Figure 1), our attempt at addressing this issue through the use of information visualization. TrustNeighborhoods is a visualization tool for graphical representation of document trust relationships in large-scale distributed file sharing systems. Based on the model of human trust presented in Ben Shneiderman's treatment of "circles of relationship" [21], TrustNeighborhoods uses the metaphor of the network being represented by a city with individual files visualized as buildings and organized into geographical regions bearing close and intuitive connotations to trust: your home, street, neighborhood, city, and finally the surrounding world. Each region of the city is color-coded and visualized as a concentric ring centered around the user's own house; files are automatically laid out in each ring with their size, color and orientation conveying information about the assigned and average trust of each file. The 2D mode of the visualization allows for interaction with the trust model, including dragging-anddropping files and users to change their trust levels, as well as radial grid distortions for the purpose of studying specific parts of the dataset. The 3D mode, on the other hand, affords overview and navigation of the distributed file system, drawing each file as a three-dimensional building on a city grid and supporting the user metaphor of trust in the context of a city. The 3D mode also has a special "fly to" command that smoothly animates the view from the user's personal house circle to the position of the requested file, showing its position in the trust relationship as well as its immediate neighborhood.

We have developed a prototype implementation of the TrustNeighborhoods visualization technique using a simulated distributed file sharing system; the input data is static and derived from an XML file, but the interface with the data source is exchangeable with a real file sharing implementation. While we have yet to conduct a formal user study of our system, informal testing indicates that the technique is intuitive and easy to learn. Subjects were able to make use of the city metaphor with a minimum of instruction and agreed that it would greatly ease understanding trust and security in a distributed file system.

In this paper, we first describe the related work, giving both a background in distributed file sharing systems on the Internet as well as the concept of trust and how to visualize it. We then describe Shneiderman's circles of relationship, and our adaptation of the model to a city metaphor. We close the paper with a discussion and some conclusions.

2 Related Work

There exists a wide array of distributed file sharing systems on the Internet today; examples include Freenet [4], Napster [17], Gnutella [7], Kazaa [10], BitTorrent [2], and so on (see for example Saroiu et al [9] for an overview). File sharing has gone from being an activity used by a select few to a mainstream Internet service enjoyed by a wide variety of users, both experts and novices; Kazaa, the most popular system today, has an average of 3 million concurrent users online at any given time. However, both cognitive and technical aspects of security need to be addressed for these kinds of systems. In regards to the technical side, most current security resides in client-side firewalls and anti-virus software on the users' local computers. Thus, current file sharing security is reactive rather than proactive. In contrast, the work presented in this paper attempts to help even novice and intermediate users (i.e. the user groups most likely to be negatively affected by malicious files) understand the relative danger of downloading specific files and avoiding security breaches before they occur by employing cognitive aids.

Another aspect of computer security that is often overlooked is the human factors viewpoint; even if a program is secure, security may still fail if it is used improperly. Whitten and Tyger [25] notes that more than 90% of all computer security failures happen due to configuration errors, facts that indicate that security is inescapably an user interface design problem. They go on to analyze the PGP privacy software to point out examples of inadequate design that may provoke users to perform fatal mistakes, such as sending unencrypted messages or divulging private information. Yee [27] argues that usability and security need not conflict, and presents a number of general design principles for designing secure and usable software. For file sharing systems, studies show that users of such systems often have difficulties understanding which of their own files are shared and which are not, and many unwittingly share personal or private files on the public network [8]. Our work is an application of these ideas, attempting to provide a usable visualization of security to facilitate secure file sharing.

The security concept we employ in this treatment is the concept of *trust* (see

Section 3.2 for a formal definition of this concept). Going into a detailed description of related work in this field is outside the scope of this paper; please refer to Marsh [14] for an exhaustive list of references to other work in the field. Rather, in this paper, we concern ourselves with the visualization of trust using techniques from the information visualization area. Not many prior examples of trust visualizations exist; the most straightforward approach is to show trust as a node-link diagram, one such example being the visualization of decentralized "webs of trust" for PGP keys using the sig2dot [22] and similar programs.

The TrustNeighborhoods visualization draws influences from a number of sources; the technique is a radial space-filling (RSF) one akin to [1, 3, 24], but it is not used for hierarchy visualization (like in the cited cases) and thus does not possess the parent-child property for circle arcs of classic RSF techniques. The interaction technique for radial distortion of society rings is a focus+context [6] technique similar to that of the InterRing [26] system, also an RSF technique. Many examples of hierarchy visualizations that could be used for file systems, both local and distributed, exist; for example, the classic cone trees [19], hyperbolic layouts [12, 13, 16], and botanical tree visualizations [11], to name just a few (see [23] for a survey). However, objects in distributed file sharing networks are typically organized in flat and shallow hierarchies (see Section 3.3), and thus the focus of the TrustNeighborhoods technique lies not in scalable hierarchy visualization, but rather on the cognitive aspects relating to security and trust. Note that the latter of the papers mentioned above happens to be of special interest due to the parallels of using a real-world metaphor for visualization: cities for our work, and botanical trees for the cited work.

3 Background

The TrustNeighborhoods visualization technique builds on two fundamental concepts: Shneiderman's "circles of relationship" and trust in network security. In this section, we give a brief background on both of these concepts. We also give a simple model for generic distributed file sharing networks.

3.1 Circles of Relationship

In his book, *Leonardo's Laptop* [21], Ben Shneiderman describes his quest for finding a suitable model for human needs in relation to computer usage, studying such ideas as Maslow's hierarchy of human needs [15] and the U.S. Declaration of Independence. The model Shneiderman finally arrives at he calls *circles of relationship* and is a simple concentric stack of circles describing human relations (see Figure 2). Each circle represents a specific class of relationship in terms of trust, shared knowledge, and personal contact, and the hierarchy starts with your own self, and proceeds outwards to circles of weaker relationships. In the same book, Shneiderman also formulates the *four stages of human activity* (collect, relate, create, and donate) which are used to separate out the activities that users participate in. He argues that these two concepts can be used to model human needs for computer usage and form a taxonomy for classifying existing applications as well as identifying potential new applications.

In this work, we adapt the concept behind Shneiderman's circles of relationship to a real visualization instead of merely using the taxonomy as a meta-framework for classification. Relationships are equalized with trust, and are made even more concrete by connecting them to a city metaphor and the various parts of a city that bear close connotations to trust. Instead of using four fixed circles of relationship, we generalize this to allow for any monotonically decreasing sequence of trust levels, each called a *society*. See Section 4 for details.

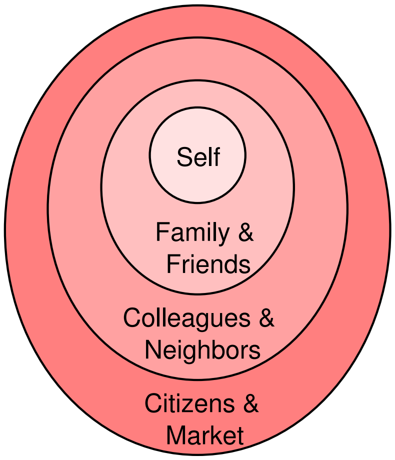


Figure 2: Circles of relationship.

3.2 Trust and Security

The concept of *trust* is used in a wide number of disciplines (see [14] for a good overview) and its definition varies with the application area. In this paper, we assume the definition of trust used in network security, where trust is the ability of one party to be able to rely on the actions or information of another party. More specifically, this amounts to reliance on the availability, integrity, and confidentiality of the other party.

In our treatment of trust, we use the concept of *trust levels*, i.e. the ability for a user to trust (or distrust) another user to a certain degree. Using the formalism of [14], we denote that a user x trusts a user y as $T_x(y)$. This measure of trust takes on values found in the range $[0, +1)$[1]. Positive values signify trust and zero values indicate impartialness. Trust can be propagated but not transferred. Moreover, trust is not symmetric, nor transitive; $T_x(y)$ has no relation to $T_y(x)$, nor do two trust values $T_x(y)$ and $T_y(z)$ say anything about $T_x(z)$.

Even if trust is not transitive, it is sometimes useful to use the judgment of others when dealing with parties we have yet to form an opinion about. For this purpose, we use the concept of *average trust* $\overline{T}(y)$ for a user y, defined as

$$\overline{T}(y) = \frac{1}{|A|} \sum_{a \in A} T_a(y)$$

where A is the set of users with non-zero trust in y. We can optionally weight $\overline{T}(x)$ with our own trust $T_x(a)$:

$$\overline{T}_x(y) = \frac{1}{|A|} \sum_{a \in A} T_a(x) T_x(a)$$

These measures we can then use when assigning trust to documents owned by another party. The exact assignment depends on the trust management policy we are using; for instance, the trust of user x in a document D_y owned by user y could

[1] Marsh uses the range $[-1, +1)$ in his formalism, where negative values signify distrust. We have no need for this added expressiveness in our work.

be defined simply as $T_x(y, D_y) = \max(T_x(y) - T_y(D_y), 0)$, where $T_y(D_y)$ is y's own trust classification the document.

3.3 Distributed File Sharing Networks

Our approach for trust visualization is not associated with any specific real-world file sharing system, but instead we use a simple theoretical model for such a system for the sake of genericity. A distributed file sharing system is typically an Internet service with two primary modes of operation: object lookup and object transfer. Object lookup can either be done through a central lookup server, as is the case for the Napster network, or through decentralized peer-to-peer (P2P) techniques, such as Gnutella. Objects are owned by specific peers, and the object lookup operation normally returns a flat list of object meta-data, including name, size, and location. In addition, we also associate our own trust with the document (0 if we are indifferent or have no previous knowledge of the entity), as well as the owner's trust in the document. Object transfer is then trivially performed by directly contacting the object owner. The TrustNeighborhoods technique is tailored at this model, but can be modified to support classic hierarchical distributed file systems by recursively coalescing objects belonging to a specific folder into compound graphical entities.

4 Trust Visualization

In our adaptation of Shneiderman's circles of relationship, we map his ideas to an even more tangible metaphor: a city used as a model for trust relationships. The intuition is the geographically correlated connotations to trust that is intrinsic to the various parts of a city: the safety of your own personal house, your home street, your neighborhood, the rest of the city, and the surrounding world. Inhabitants of your house you tend to trust explicitly; neighbors in your street slightly less so, and even less for your neighborhood, the city as a whole, and the country in which your city is located. This metaphor is then used to categorize users and documents on a distributed file sharing network. According to the metaphor, documents and users you encounter in the confines of your own house are intuitively seen as highly trusted and safe, whereas documents or individuals found in the outer parts of the city can be seen as potentially "shady" and should be handled carefully and the information regarded skeptically.

Taking this a step further, we can then use this trust information to classify documents owned by other users in a distributed file sharing system. Using a combination of our own trust assignment policy and the average trust of other users, external documents are classified and added to the same city model. The final result is a dynamic and easily overviewable picture of the trust relationships in the system, giving users a way to easily relate the geographical position of a document to its trust level.

Our information visualization technique building on these ideas is called *TrustNeighborhoods*, and is a space-filling radial-layout visualization consisting of concentric rings representing the various trust levels and the buildings on the city grid representing individual documents and users in the system. To simplify the concept of trust, the trust levels are discretized into a small number of regions that we call *societies*. Each society has an associated name and color; in our implementation, the names are "house", "street", "neighborhood", "city", and "world", and the colors are green, blue, yellow, red, and grey, respectively (see Figure 3 for an example). These names serve to connect the model more strongly to the city metaphor, while the colors are selected to give some indication of the trust level of each society (green being perceived as a “safe” color in Western cultures, while red signifies danger, and the outmost grey color for zero trust represents indifference).

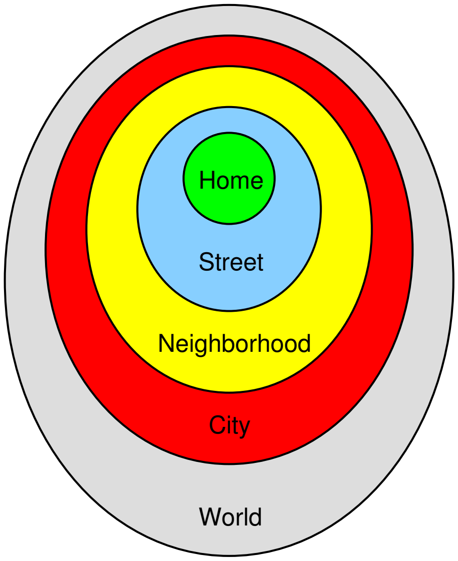


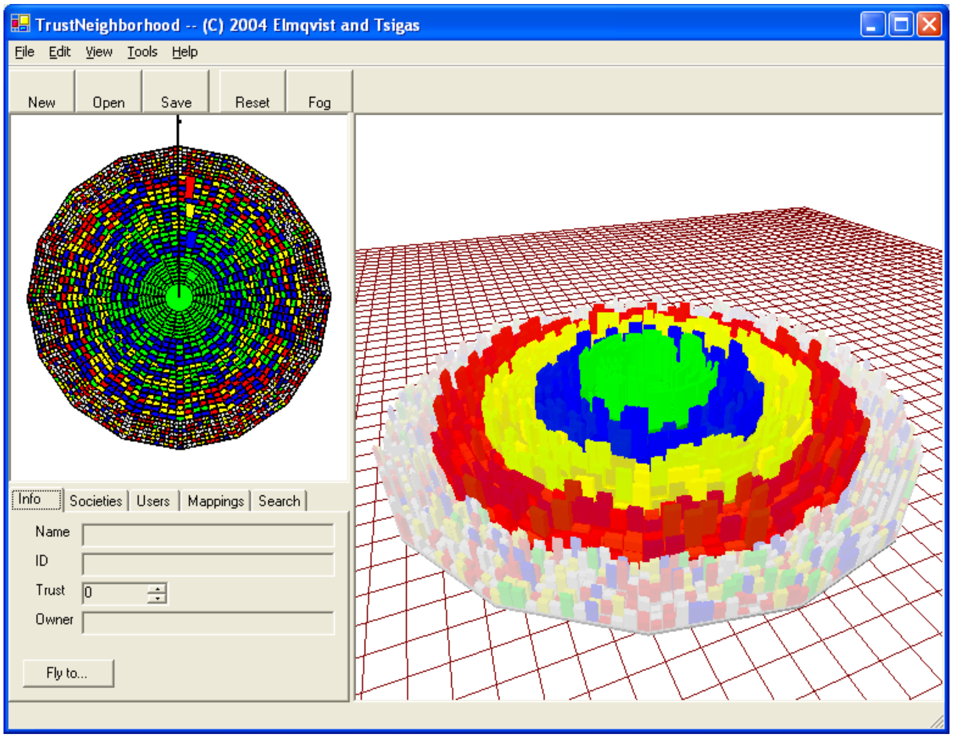
Figure 3: Standard TrustNeighborhoods societies.

The technique can be used to visualize the trust relationships of both users as well as documents in the system. In both cases, the entities are organized into societies according to their trust level. They are then placed in the radial grid of each society ring. Angular placement is controlled by the user to allow for spatial arrangement that makes sense to the user, but could optionally be sorted according to some system property. See Section 4.5 for more details.

Finally, the visualization itself uses this representation of societies consisting of stacked ring arrays to render a graphical image of the system. The TrustNeighborhoods method has both a 2D and a 3D mode using the same representation; the 2D mode is intended for managing trust and societies, whereas the 3D mode is used for overview and navigation. The behavior of the graphical visualization depends on the mapping between system properties and the graphical actuators; for instance, the user can configure the visualization to map the average trust of a document as the height of the city building representing it. Section 4.3 describes the mappings between properties and actuators in TrustNeighborhoods. Figure 4 shows and example of both a 2D and a 3D a Neighborhoods document visualization of some 3000 documents belonging to 100 different users in a distributed file sharing system.

4.1 2D Mode: Trust Management

In the 2D mode of the TrustNeighborhoods method, we simply draw the societies as concentric rings, using the identifying color of each society as a background. Entities are then drawn in a second pass as circle segments on top of the ring representing the society. If we are visualizing documents, we can draw the circle segment for a document D_y in the color representing either simply the owner's y own trust classification (i.e. we use $T_x(D_y) = T_y(D_y)$), or we can alternatively weight this value with our own trust in the user (i.e. $T_x(D_y) = T_x(y)T_y(D_y)$). If, on the other hand, we are visualizing users instead of documents, a useful metric for the color of the circle segment is the average trust of the user y in the system, possibly weighted by our own trust ($\overline{T}(y)$ or $\overline{T}_x(y)$, respectively, see Section 3.2).



Trust 0

Figure 4: TrustNeighborhoods visualization for 100 users and 3000 files.

The mapping between system property and actuator ("color" being the actuator in this particular case) is entirely up to the user. Regardless of which property is used, the entity color is designed to give extra information to the local user, for instance highlighting asymmetric relationships or improper trust assignment ("I trust that user, yet he is not particularly well-trusted among my peers–maybe I should revise my trust?").

Note that societies are disjoint intervals on the trust range $[0, +1)$; finding the appropriate society and its corresponding color is as simple as finding the interval containing the calculated trust value.

As stated above, the primary purpose of the 2D mode is trust management for both users and documents, giving users an intuitive control over the trust relationships in the system. For user trust management, the user can simply drag-and-drop other users in the society hierarchy, changing their trust level. Users with zero trust, i.e. those users we have yet to classify, simply do not show up in the Neighborhoods visualization. This is done for scalability reasons, since the total number of users in the system could be very large. Only when the local user searches for other users with some wildcard will the world society (trust level 0) be populated with the search results.

For document trust management, on the other hand, the local user can control the trust levels of the documents she herself owns, but cannot affect the trust level of other documents. She can, however, change the position of the non-local document within the society ring to allow for spatial arrangements that make sense to the user. Changing the trust level of a user will indirectly change the trust level of the documents owned by that user.

Our implementation of the 2D TrustNeighborhoods technique also supports continuous zooming and panning in the visualization to simplify trust management for complex systems. Users can easily zoom in and out of the visualization and pan

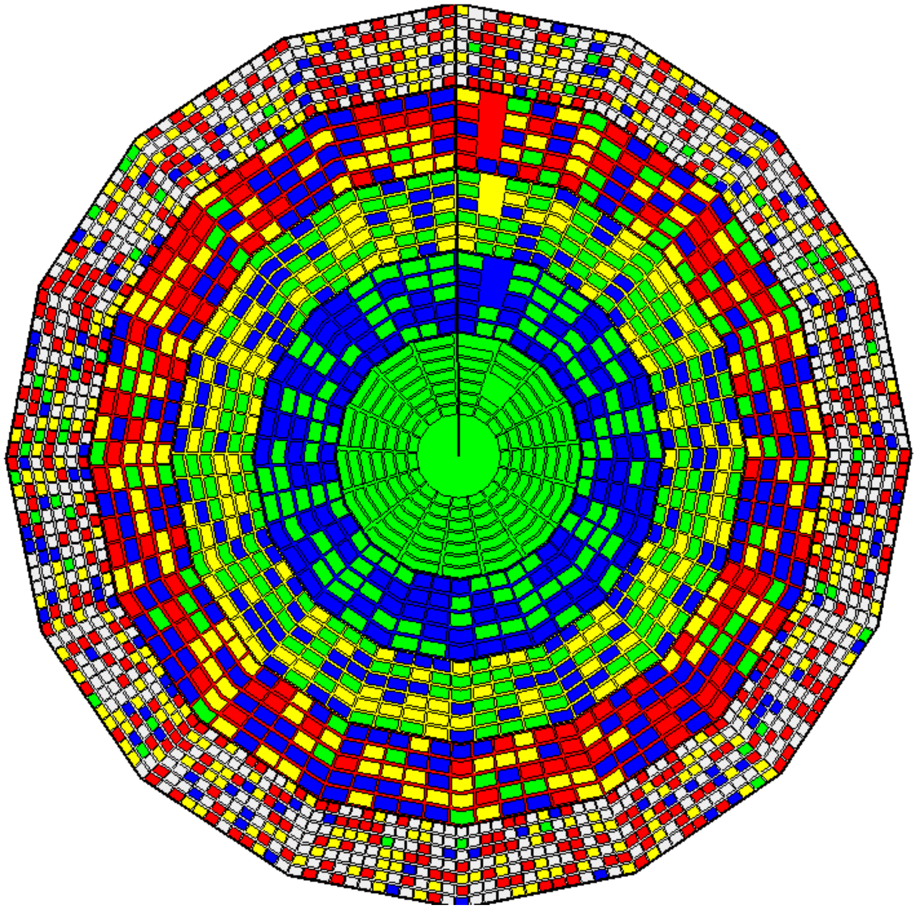


Figure 5: TrustNeighborhoods 2D visualization.

around to study details in the system being visualized.

4.2 3D Mode: Overview and Navigation

The 3D mode of the TrustNeighborhoods technique is primarily designed for overview and navigation. The idea behind this mode is for it to act as a mental aid to a local user accessing documents on the file sharing service; whenever the user is downloading a document, the visualization will pop up to show the context of the document being downloaded and give the user a feel for its trust. In other words, the 3D mode is not intended to serve as a file browser or search interface (even if it can be used that way). Instead, the local user would spend a small amount of initial effort to build up a trust relationship using the 2D mode, and then entrust the system to compute trust appropriately and making it explicit to the user through the 3D mode. Only seldomly would the user have to go back to the 2D mode to perform trust management maintenance.

The heart of the 3D TrustNeighborhoods visualization is again the stack of societies representing trust levels and the categorization of documents and users into these, but here the metaphor of a city is much stronger than for the 2D mode. The whole data set is rendered on a set of color-coded concentric circles representing the society rings, each ring slightly taller than the one outside it. Entities (i.e. documents or users, depending on which mode the visualization is in) are then rendered as buildings in their respective society rings, mirroring their placement in the 2D version of the visualization. Thus, the user is able to easily recognize the trust level of a specific entity by observing its location in the city, an operation that would require only a cursory glance at the visualization window.

Each building has a number of attributes that are connected to actuators for the visualization; see the next section for more details. These attributes control the graphical look of individual buildings and can thus carry information to the user of varying degrees of subtlety; in our prototype implementation, we use the building color, height, and roof angle, but more attributes such as shape, texture, and size are certainly feasible.

To further aid users in quickly assessing the trust of specific entities, the visualization also supports the use of volumetric fog; buildings in each society ring are shrouded in volumetric fog with the same color as the society itself. This feature gives a visual indication of the trust level of a society at a distance, but still affords exact color recognition when the camera draws closer. Nevertheless, fog can toggled on and off depending on the desires of the user.

4.3 Properties and Actuators

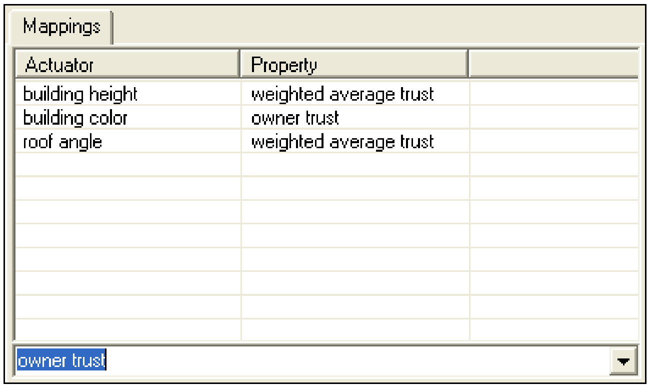
In the interest of flexibility, the Neighborhoods visualization has configurable mappings between the *entity properties* for users or documents and the *graphical actuators* that actually carry information in the visualization (both 2D and 3D modes). Only the assignment of entities to specific society rings depending on their trust level remains invariant; all other properties can be mapped freely to the available actuators to fit the needs of the user. Tables 1 and 2 give the pre-defined actuators and properties of our implementation of the TrustNeighborhoods technique. The mapping configuration window (see Figure 6 for an example) allows the user to control the mapping explicitly. In the example, the user has connected the height of 3D buildings to the weighted average trust of the document, the roof angle signifies owner trust, and the color the document size. The number of actuators and properties can naturally be extended with additional factors.

Actuator	**Description**	**Mode**
roof	angle of building roof (45° to 0°)	3D
height	building height	3D
color	building color	2D/3D

Table 1: Graphical actuators for the TrustNeighborhoods visualization (both 2D and 3D).

Property	**Description**	**Type**
size	document size (in bytes)	doc
owner trust	document owner trust ($T_y(D_y)$))	doc
w. owner trust	weighted owner trust ($T_y(D_y)T_x(y)$)	doc
avg. trust	average trust ($\overline{T}(y)$)	doc/usr
w. avg. trust	weighted average trust ($\overline{T}_x(y)$)	doc/usr

Table 2: Standard entity properties for the TrustNeighborhoods visualization.



Actuator	Property
building height	weighted average trust
building color	owner trust
roof angle	weighted average trust

owner trust

Figure 6: Mapping configuration window for connecting actuators to properties.

4.4 Searching and Pruning

Attempting to visualize all objects in a distributed file sharing system would be folly, and a majority of the objects are of no relevance to the user anyway. In the TrustNeighborhoods technique, we visualize all files that belong to users that the user has a non-zero trust rating for, as well as files that match the current search criteria.

In addition to searching, the visualization technique also supports entity *pruning* to cut down on the amount of concurrently visulized objects. Pruning criteria, unlike search criteria, affect all active entities, and filters out entities that do not match the desired properties. This can include file size, trust, average trust, or variations and combinations of these.

4.5 Structure and Layout

Entities (i.e. files or users) visualized in the TrustNeighborhoods technique are arranged radially according to their trust level. However, in each society ring for a particular trust level, there is no specific layout mechanism. The space is split into a grid according to an algorithm given below, and entities are simply added to the grid in arbitrary (but stable) order. This means that the user is free to rearrange the placement of individual entities in a society according to her liking, i.e. to group together certain files that have a semantic connection, for instance.

The available space in each society ring is split into a rectangular array of length n (wraps around) and width r, sized to fit all entities contained in the society. A second criteria is added for aesthetical reasons; n and r are chosen so that the inner edge of the innermost elements have an edge dc equal to their radial edge dr (see Figure 7). Documents (or users) are then assigned a specific position in this radial array; in the case of the array having to be resized to accommodate more or fewer entities, the data structure supports a dynamic resizing operation which tries to maintain the relative position of each element as far as possible.

Sometimes it might make sense to impose some specific order on entities in a society ring. To support this, the TrustNeighborhoods technique also supports sorting of entities according to one of the system properties (i.e. average trust, owner trust, file size, etc). Sorting operations are confined to the currently selected society ring. To give users more control over the results of the sorting, the 2D mode of the visualization has a “dial” (see Figure 8) that is used to specify where the first of the entities in the sorted list should be placed.

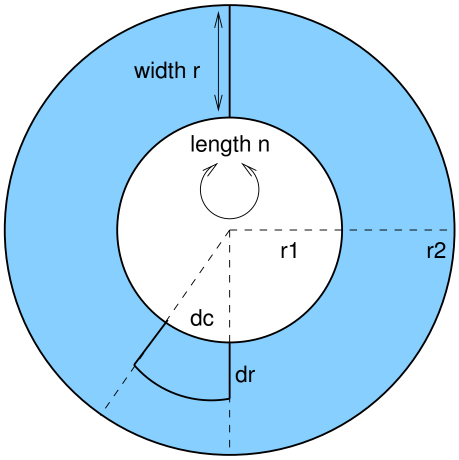


Figure 7: Society ring layout strategy.

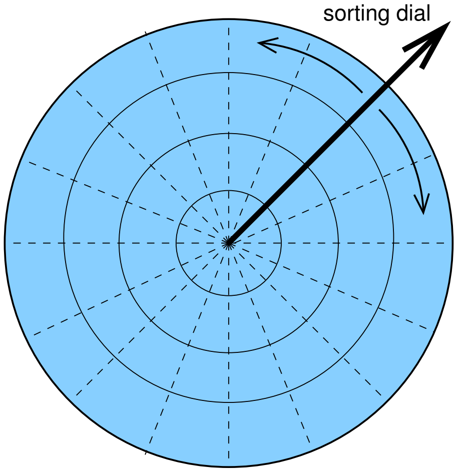


Figure 8: Dial (arrow) used for controlling the start point of the sorting operation.

4.6 Interaction Techniques

In addition to the interactions specified above, the TrustNeighborhoods visualization also supports two additional direct manipulation [20] techniques which are used for interacting with the visualization: a mechanism for modifying the visual space assigned to the various society rings, and a "fly to" command that is used to transport users from an overview to a detail view in the 3D mode of the visualization.

4.6.1 Society Distortion

Users are likely to assign different importance to different trust levels in the society hierarchy, and thus we provide an interaction technique that allows for control of the visual space allocated to the various society rings. This technique is similar to the radial distortion described in [26] or the linear time windows [5], and allows for turning the TrustNeighborhoods visualization into a focus+context [6] technique; the visualization can be focused on a certain part of the object hierarchy, with the surrounding society rings providing context. Actually performing the radial distortion is either done in a direct manipulation fashion by clicking and dragging in the 2D mode of the visualization, or calling up the relevant dialog box and changing the area ratios between the rings there.

4.6.2 Fly To

The central interaction for the main intended use of the 3D mode of the visualization is the "fly to" command that transports the user from an overview of the whole city down to a detail view of a specific building in the city and its immediate neighbors. Recall that buildings represent either a document or a user. The purpose of this command is thus to provide the user with the context of a specific file she is downloading or a remote user she is interacting with in relation to the rest of the network that the whole city visualization represents.

4.7 Implementation

We have developed a prototype implementation of the TrustNeighborhoods visualization technique using the C# programming language on the Windows platform. The application (see Figure 1 and 4 for screenshots) uses the standard Microsoft .NET SDK and the CsGL OpenGL bindings for C# for the graphical rendering of the visualization techniques. Instead of interfacing with a real distributed file sharing system, the application accepts a generic XML description of the current users and files in the system and uses this for visualization. This means that the data set is static rather than dynamic, but the application has been designed with possible modifications to using a real data source in mind.

4.8 User Study

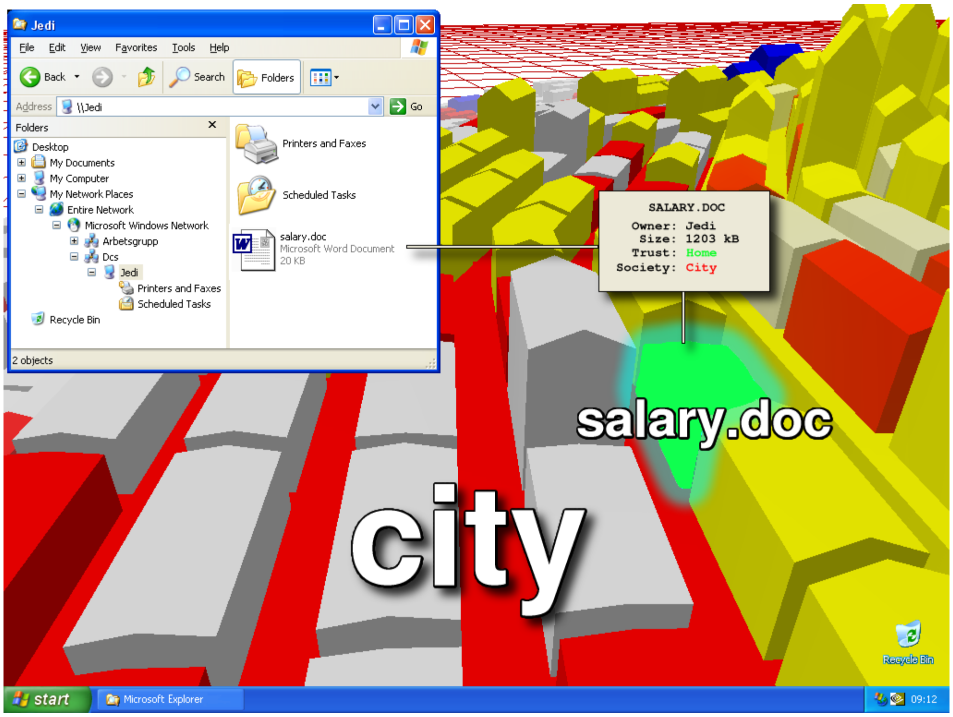
We conducted an informal user study of our implementation of the TrustNeighborhoods technique with a number of our fellow colleagues as well as a pair of high school students with only basic computer training. While not strictly the target audience for this visualization, the test subjects agreed that the tool was both novel and intuitive. Understanding the basic idea behind the city metaphor was quick and easy, and even expert subjects claimed that they could make good use of such a visualization for understanding file system trust.

5 User Environment Integration

Our vision for the use of TrustNeightborhoods is to integrate it with existing user desktop environments such as Microsoft Windows and MacOS. The user would be able to assign trust to just a few known users with the help of a standard interface for the distributed file sharing mechanism. Whenever the user requests access to a remote file on the network, the "fly-to" interaction of the 3D view could be invoked in the background of the desktop environment, non-obtrusively providing the user with the trust context of the file. Figure 9 shows a mockup screenshot of how this system could look when integrated with Microsoft Windows XP.

6 Discussion

One of the primary concerns of almost any information visualization technique is whether it scales appropriately or not. In the case of distributed file sharing networks, we are potentially dealing with thousands upon thousands of users and millions of files. It is clearly infeasible to try to visualize all of the files or the users in such a system as individual entities. On the other hand, it is also not necessary; the user will only be interested in a select few remote users and their files, and



SALARY.DOC

Owner: Jedi
Size: 1203 kB
Trust: Home
Society: City

Figure 9: Mockup of integrating the TrustNeighborhood 3D visualization with the Microsoft Windows XP desktop environment. (Note that this is a composite screenshot.)

will have no need to see the whole network. In the TrustNeighborhoods approach, we employ various pruning mechanisms to solve this problem. Users that we have zero trust in are not interesting for the local user, and are thus never displayed; the absolute majority of users in the file sharing system will be of this category. Documents belonging to a user with a non-zero trust, but which has a modified local trust of zero, are also not interesting, and may optionally be pruned as well. Only in the event of the local user explicitly searching for a user or a document with zero trust will we include these in the visualization (essentially using the outermost "world" society as a container for search results).

Furthermore, the radial space-filling layout used in the TrustNeighborhoods technique is fairly scalable by itself. Societies adapt seamlessly to the number of entities contained in them, and split the available space in appropriate-sized segments. Also, the continuous zooming mechanisms present in both the 2D and 3D modes of the method should improve its scalability. Figure 4 shows a fairly complex example of a system of a 100 separate users and some 3000 files. Going on our informal testing of the system, our belief is that the TrustNeighborhoods technique scales appropriately for its application area, even if we have yet to connect the visualization to a real file sharing system.

Another issue that bears some discussion is the choice of using colors to represent trust. The color selection used in the reference implementation is based on "traditional" Western color perception, where green and blue is seen safe and friendly, whereas yellow and red signifies danger. The fact is that different colors bear different connotations in different cultures. This also makes the combination of different colors to show gradients of trust (instead of discrete trust levels) slightly more troublesome. In the TrustNeighborhoods implementation, we support both of these concepts through special options, allowing the user to configure the colors used for various trust levels, as well as supporting color gradients.

No visualization that makes direct use of Shneiderman's circles of relationship would be complete without at least a brief discussion on where in the activites and relationships table (ART) the application itself should be placed. While the straightforward answer is that TrustNeighborhoods is an information collection activity for your own personal use, the distributed file sharing system it is designed to interface with could potentially be used for all of the four stages of human activities and involving all of the four circles of relationships.

7 Conclusions

We have presented TrustNeighborhoods, an information visualization technique using Shneiderman's circles of relationships implemented using a city metaphor to show the trust level of documents in a distributed file sharing system by relating them to their geographical position in the city. The technique is aimed at novice and intermediate-level users, giving them an intuitive way to quickly see the trust and surrounding context of documents they are accessing on the network using a built-in "fly to" command in the 3D mode of the visualization. We have implemented a prototype of the technique using a dummy XML data source, and our informal testing indicates that the method is both useful and easy to grasp. To summarize, the main contributions of this paper are the following:

- an adaptation of Shneiderman's circles of relationship to a tangible city metaphor useful for information visualization;
- an intuitive, flexible, and accessible technique for trust visualization building on this city metaphor;
- interaction techniques for manipulating the trust hierarchy, managing individual entities, and automatically navigating to a specific location in the 3D environment;
- dynamic layout and sorting strategies for quickly arranging a whole group of entities according to domain-relevant criteria in a way that conforms to the city metaphor; and
- a strategy for how to integrate this visualization technique with a real user environment in order to non-obtrusively display a visual 3D indication of the trust level of a remote file being accessed.

8 Future Work

A possible improvement to the TrustNeighborhoods method would be to project the city grid on top of a texture-mapped 3D terrain mesh to further utilize human 3D spatial cognition, similar to the Data Mountain [18] technique. In fact, it might be worthwhile to emphasize making the visualization look even more like a real city by allowing for customizing buildings and landmarks on the city grid, and using the street network for conveying additional information about connectivity and structure.

Acknowledgements

Thanks to the subjects who participated in the informal user study and offered valuable insight and comments on how to improve the visualization.

References

[1] Keith Andrews and Helmut Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *Proceedings of the IEEE Symposium on Information Visualization 1998 (Late-Breaking Hot Topics)*, pages 9–12, 1998.

[2] Bittorrent, 2005. *http://www.bittorrent.com.*

[3] Mei Chuah. Dynamic aggregation with circular visual designs. In *Proceedings of the IEEE Symposium on Information Visualization 1998*, pages 35–43, 1998.

[4] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46–67, 2001.

[5] Niklas Elmqvist and Philippas Tsigas. Citewiz: A tool for the visualization of scientific citation networks. Technical Report CS:2004-05, Department of Computing Science, Chalmers University of Technology, 2004.

[6] George W. Furnas. Generalized fisheye views. In *Proceedings of the ACM CHI'86 Conference on Human Factors in Computer Systems*, pages 16–23, 1986.

[7] Gnutella, 2005. *http://www.gnutella.org.*

[8] Nathaniel S. Good and Aaron Krekelberg. Usability and privacy: a study of Kazaa P2P file-sharing. In *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, volume 1 of *Privacy and trust*, pages 137–144, 2003.

[9] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking*, 2002.

[10] Kazaa, 2005. *http://www.kazaa.com.*

[11] Ernst Kleiberg, Huub van de Wetering, and Jarke J. van Wijk. Botanical visualization of huge hierarchies. In *Proceedings of the IEEE Symposium on Information Visualization 2001*, pages 87–96, 2001.

[12] John Lamping and Ramana Rao. The hyperbolic browser: A focus + context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing*, 7(1):33–35, 1996.

[13] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems*, 1995.

[14] Stephen Paul Marsh. *Formalising Trust as a Computational Concept.* PhD thesis, Department of Computer Science and Mathematics, University of Stirling, 1994.

[15] Abraham Maslow. *Towards a Psychology of Being.* Wiley, 1998.

[16] Tamara Munzner and Paul Burchard. Visualizing the structure of the world wide web in 3D hyperbolic space. In *Proceedings of the Symposium on Virtual Reality Modeling Language 1995*, pages 33–38, 1995.

[17] Napster, 2005. *http://www.napster.com.*

[18] George G. Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. Data mountain: Using spatial memory for document management. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 153–162, 1998.

[19] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proceedings of the ACM CHI'91 Conference on Human Factors in Computing Systems*, 1991.

[20] Ben Shneiderman. Direct manipulation: A step beyond programming languages. *IEEE Computer*, 16(8):57–69, August 1983.

[21] Ben Shneiderman. *Leonardo's Laptop: human needs and the new computing technologies.* MIT Press, 2002.

[22] sig2dot, 2005. *http://www.chaosreigns.com/code/sig2dot/.*

[23] John Stasko, Richard Catrambone, Mark Guzdial, and Kevin Mcdonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International Journal of Human-Computer Studies*, 53(5):663–694, 2000.

[24] John Stasko and Eugene Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Visualization 2000*, pages 57–68, 2000.

[25] Alma Whitten and J. D. Tygar. Why Johnny can't encrypt; A usability evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium*, pages 169–184, Berkely, CA, 1999. Usenix Association.

[26] Jing Yang, Matthew O. Ward, and Elke A. Rundensteiner. InterRing: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceeedings of the IEEE Symposium of Information Visualization 2002*, pages 77–84, 2002.

[27] Ka-Ping Yee. User interaction design for secure systems. In *International Conference on Information and Communications Security (ICIS)*, 2002.