Strata: Physical Representations for Layered Information Structures

Brygg Ullmer, James Patten, and Hiroshi Ishii
Tangible Media Group
MIT Media Lab

Abstract

The Strata project explores the design of layered, computationally-mediated physical models that embody user interfaces for specific layered information structures. The interface is constructed of an array of flat-panel displays with digitizer input, and a series of physical tokens embodying parameterized SQL queries to an underlying relational database. Strata's first application is a network and facilities management interface, represented in a model embodying the physical structure of a laboratory building. We present the motivation, design, and implementation of Strata, and discuss continuing and future work.

CR Categories and Subject Descriptors: H.5.2 [User Interfaces] Input devices and strategies; H.5.1 [Multimedia Information Systems] Artificial, augmented, and virtual realities

Additional Keywords: tangible user interface, tangible bits, physical constraints, augmented reality, information visualization

1 INTRODUCTION

The challenge of information representation is one of the most fundamental issues underlying how people communicate with each other, reason about our world, and interact with our increasingly mechanized tools and systems. Traditionally, people have relied primarily upon *visual representations* – esp., text and graphics – to interact with computational systems. The combination of dynamic visual representations with general purpose input devices has yielded flexible systems equally suitable for spreadsheets and data visualization, graphic design and web browsing.

A growing stream of research has begun to explore an alternate approach called "tangible user interfaces" [4], where *physical representations* are used to mediate interaction with computational systems. By way of analogy, traditional games like chess, poker, and monopoly use diverse systems of physical objects to "embody" abstract rules and physically "mediate" human interactions. Like the abacus of [4], these systems make no distinction between "input" and "output," or between representation and control. Instead, component artifacts serve simultaneously as embodied representations of system state, as well as physical controls for directly manipulating their underlying associations.

As a relatively young area of research, many basic issues of tangible interfaces remain poorly understood. For instance, which aspects of a computational system can profitably be physically represented, and how? What kinds of fusions between physical and visual representations are possible? Perhaps most simply put, what should be physical, and what should be digital?

One approach is to consider the kinds of entities which can be computationally represented. For instance, the "Urp" urban planning system of [12] – perhaps the most sophisticated tangible interface to date – integrates physical representations of information elements (e.g., individual buildings), functions (e.g., wind simulation), and operators (e.g., wind probe and material wand). Each physical object has a direct mapping to a single computational association (*a* building, *a* function, etc.). While Urp supports interaction between multiple tokens (e.g., the application of the material wand to a building), the aggregation of multiple objects into higher-order information structures is not supported.

The mediaBlocks system [10] made progress in supporting physical interaction with higher-order information structures. The system combined blocks associated with lists of media elements (images, video, etc.) with physical constraint structures that operate on one or more of these list-objects. In the process, mediaBlocks illustrated not just a system for media manipulation, but the beginnings of a broader repertoire of interaction techniques useful for many interactive systems employing list-like elements.

Building on this earlier work, we will introduce a tangible interface providing physical representations and controls for more complex information structures. In particular, we will present Strata, a tangible interface for monitoring and manipulating the networking and facilities infrastructure of our laboratory's building – a complex system characterized by a series of interrelated physical and logical layers.

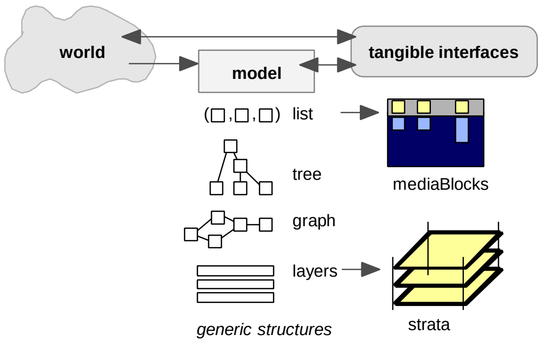

Figure 1: Physical embodiment of information structures

2 STRATA

In the Strata project, we have developed a tangible interface for the representation and manipulation of layered information structures. A wide variety of real-world systems fall into this general category. As the project name implies, many geological, atmospheric, biological, and other naturally occurring systems are characterized by layered physical structures. Many man-made systems are also layered in physical structure, from the macroscale of cities to the microscale of integrated circuits, and beyond.

Systems may display both physical and logical stratification. For instance, archaeological sites are often considered in terms of physical layers reflecting different time periods. Alternately, while geographical information systems (GIS) are often used to relate different kinds of natural and man-made processes sharing a common geographical reference frame, some of which may be of purely conceptual nature – e.g., urban zoning parameters.

Conversely, both physical and logical systems can display stratification. Where most of the above examples illustrate physical systems, businesses org charts depict administrative and functional stratification within organizations, and computer networks reflect layering at scales ranging from protocol stacks to the Internet. In general, many complex systems can be considered in terms of layered structures, where certain spatial, temporal, and parametric properties are held constant, and others are varied across discrete or continuous layers.

2.1 Network and facilities management

Our own effort began with an attempt to understand wireless network traffic patterns across the physical rooms and floors of our laboratory building. We discovered that by monitoring the network traffic of mobile computers shifting across the building's wireless transponder zones, we could infer the beginning and character of various meetings, and correlate these to scheduled activities on the building's calendar system.

To aid the understanding of this information, a 3D graphical model of the building was constructed. Wireless activity from our router traffic logs was displayed with color-modulated nodes, spatially positioned within transparent layered texture-maps of the buildings floorplan (Figure 2). This 3D graphical representation had an unexpected outcome: correlations were observed between similarly-positioned network access points on different floors, possibly reflecting cross-floor wireless interference. This effect had not been previously observed in the building, and would have been difficult to detect without the 3D visualization.

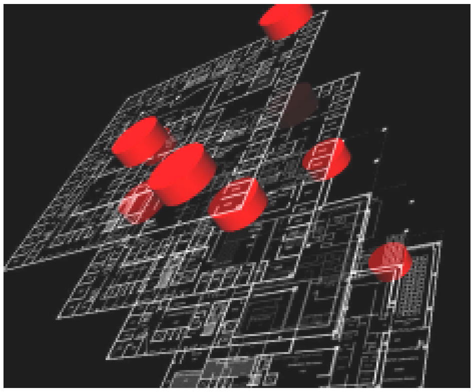

Figure 2: Early 3D graphical model of building wireless traffic

Encouraged by these early results, additional data reflecting mailing list traffic, group office occupancies, etc. were imported into the visualization. While the results were visually interesting, several problems surfaced. First, the contents of this 3D visualization could be difficult to understand, orient, and modify, especially when populated with large numbers of information nodes.

Secondly, the spatial properties of many important elements – e.g., the location of individual machines – was not available in recorded form. This appeared to be a consequence of both the difficulty of expressing and maintaining this information with existing tools, and the absence of tools for usefully employing and visualizing the resulting information.

Finally, we were unsatisfied with existing approaches for expressing queries which might identify patterns and structure within and across the many possible sources of building information. While a broader problem extending beyond the bounds of our individual interface, we believed that the right tools for elegantly composing complex queries would make a qualitative difference in user interaction with our system.

With work, each of these problems could be approached and improved within the confines of traditional graphical interface techniques. However, at other levels, these kinds of problems seemed partially endemic to the limitations of purely visual representations on general purpose computers.

2.2 Physical mockup

As a tool for considering these problems with our 3D graphical interface, we constructed a simple physical model of the building with a laser cutter. Each floor of the building was represented with a laser-cut piece of acrylic, structured vertically with spacer elements, and etched with building floorplans. (Figure 3)

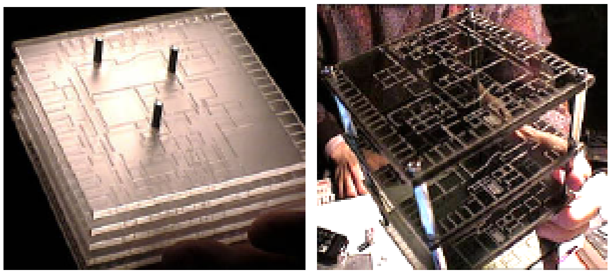

Figure 3: Uninstrumented physical models of lab building

When this model was handed to our building's facilities manager, his first response was striking. In his excitement about our project's relevance to his job, he began discussing building maintenance issues while touching associated locations in the building model – often even without looking at the physical model.

The coupling of his hard-won building knowledge, his sense of kinesthesia, and passive tactile feedback allowed him to fluidly manipulate the physical model with expertise, immediacy, and ease. Even more striking, we realized that several of the key interactions which came so readily with the physical model – e.g., using kinesthesia to manipulate without direct visual engagement – were fundamentally impossible with the purely visual representations of monitor-based 2D and 3D graphical displays.

This enthusiastic response encouraged us to migrate from our 3D graphical display to a new interface idea – that the physical model of the building could itself become the interface to the many layers of its integrated physical and digital infrastructure.

Here, an important factor visible in the passive models of Figure 3 is that the physical model accurately reflects the building as it exists in the world (or at least, the mental model the users have of this structure). Our physical interface is conceived not as a generic interface to layered buildings or information structures in the abstract, but instead as an embodiment of some *particular* structure – in this case, our laboratory building.

This aesthetic is key to realizing the kind of kinesthesia and representational fidelity key to our intended user experience. The construction of one-off or few-off models is also surprisingly practical, in light of rapid physical fabrication tools such as the laser cutter. Especially with the laser cutter, custom objects can be rendered from computer drawings in minutes or even seconds, in a fashion not unlike their laser printer kin. This facility for rapid physical manufacture is reflected throughout the physical interfaces in our paper.

3 FUNCTIONALITY OVERVIEW

The first application of the Strata interface is the monitoring and control of our laboratory computer networks. The interface is structured around two kinds of physical representations. The first of these is the layered building model, representing the physical floors of our laboratory as a series of "floor panes."

Secondly, a series of "parameter wheel" physical tokens are associated with parameterized SQL queries to an underlying relational database, and are used to access and manipulate content in the layer structure.

Parameter wheels can also be accessed through a gateway device on a GUI workstation, supporting a strong linkage between Strata and traditional network management tools.

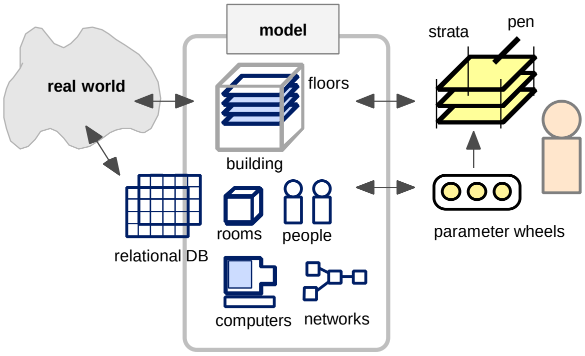

Figure 4: Conceptual model of Strata interface

3.1 Layer interface

The building's physical floors are embodied as transparent acrylic panes, each associated with a floor of the building. These panes are cut in the shape of the actual floors, and etched with the boundaries of rooms, halls, and other major building features. The panes are also embedded with wireless ID tags, so that they may be digitally identified by the interface.

The panes are used within a roughly cubic-meter aluminum frame that houses three interactive display surfaces. These flat panel displays are layered with a vertical displacement of 15cm, and mounted horizontally on drawer slides (Figure 4).

The drawer slides allow the active layers to be slid horizontally over a displacement of 50cm. This horizontal movement serves the practical function of interactively maximizing multi-floor parallelism while minimizing cross-floor occlusion. Additionally, the horizontal movement supports conceptual grouping and ordering of the layers by the user, a role we will return to in the discussion.

The interactive physical-layer displays of Strata are based upon Wacom PL400 digitizing flat panels. These support pen stylus input coincident with a 38cm-diagonal LCD display surface.

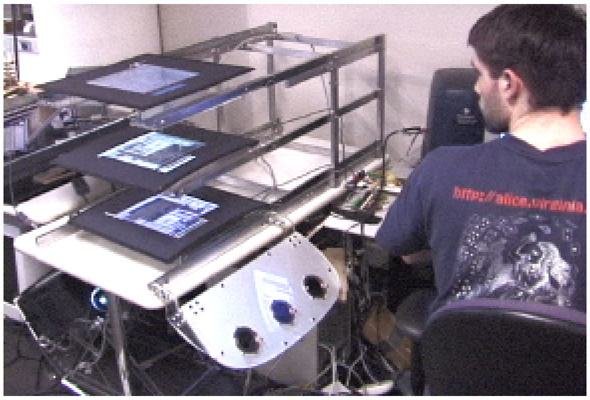

Figure 5: Strata layer interface, overview shot

These sensing/display surfaces are fronted with 40cm-square faceplates. This faceplate is inset with a 20cm-square frame which receives, senses, and graphically augments the etched acrylic building floor panes; and an adjacent but physically separated 20x5cm display surface which provides augmenting textual and other supporting information.

Information relating to building floors is accessed by placing one or more floor panes onto layers of the Strata interface. Graphical content is displayed through the transparent acrylic panes, within the context frame of the physically etched floorplans. The pen stylus supports querying and manipulation of graphical display contents, in a kind of hybrid GUI/TUI approach.

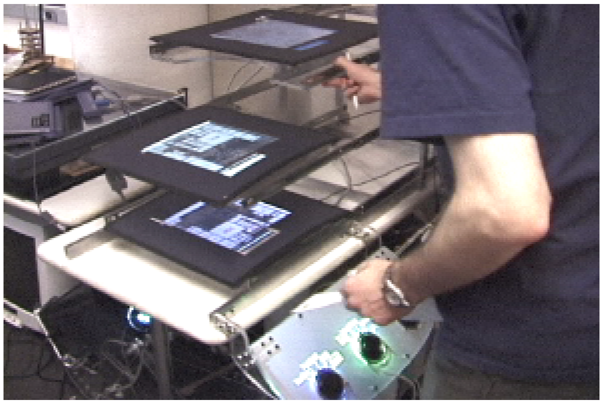

Figure 6: Strata, sliding layers

The etched floorplans provide several advantages over a purely graphical visualization. First, they provide a stable physical reference frame different in character from the transient malleability of purely graphical displays. Secondly, they provide a persistent visualization of floor plans, even in the absence of graphical mediation.

Perhaps most interestingly, the pen stylus mechanically couples with the laser-etched floorplan lines to provide a compelling form of passive haptic feedback, which can be modulated as a function of etch depth and cross-hatching. This tactile modality is used as a kind of interaction constraint and feedback mechanism.

3.2 Query interface

The floor panes do not by themselves express which of the many possible floor-specific data are to be graphically displayed. This selection is done through a series of tagged tokens that we call "parameter wheels," used within a visually mediated, mechanically constrained control panel (Figure 6).

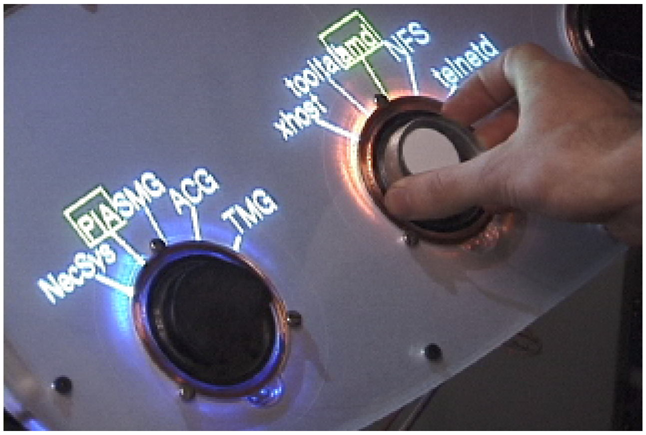

Figure 7: Strata query interface

Parameter wheels are shaped in the form of truncated discs, and are each uniquely tagged with an RF ID chip. They are activated by insertion into receptacles on the control panel. These receptacles both read the parameter wheel's digital ID, and allow the wheels to rotate clockwise or counterclockwise, which is sensed through mechanical coupling to a potentiometer.

The control panel housing these receptacles is itself a projection surface. A video projector rear-projects computer graphics onto the control panel (made of clear acrylic and vellum), allowing graphical mediations to flow up against the mechanical parameter wheel receptacle assemblies.

3.2.1 Physical embodiment of SQL relational database queries

Parameter wheels are each associated with a parameterized SQL query, which may be invoked to query and select network and facilities features in the building from a relational database. The database includes extensive crosslinked information on computers, routers, router performance logs, people, rooms, research groups, mailing list memberships, and trouble ticketing system information, among other information.

For instance, the database includes the physical location, operating system type, users, and security analyses of all computers in the building. This information is initially generated through a series of automated scripts, and is manually modifiable both within the Strata layer interface and via a suite of Web-based tools. These are described in the implementation section.

Each parameter wheel is associated with a parameterized SQL query into the database, and a list of selectable parameters. For instance, one of the most simple wheel associations selects all the rooms associated with a particular research group. SQL query results are always expressed in terms of spatial locations within the building, such that they can displayed within the Strata model.

When a parameter wheel is placed onto a control panel receptacle, its associated query parameters are graphically displayed as selectable options around the circumference of the wheel. When the wheel is turned, the selected parameter is substituted into the query. The completed SQL query is then issued, with its results displayed in the layer interface.

3.2.2 Conjunction of multiple query results

The parameter wheel control panel has receptacles for three parameter wheels. When multiple parameter wheels are simultaneously active, their query results are graphically conjoined on the layer displays in a kind of boolean "and" operation. Network nodes or offices common to the query results of multiple active parameter wheels are graphically emphasized, while results specific to a single parameter wheel are deemphasized.

This feature supports simple, rapid expression of complex interdependent queries, aiding the browsing and comprehension of the state and properties of complex systems such as computer networks.

3.3 Example usage: response to network attack

Strata has been developed with strong input from our laboratory's network and computing systems group, who are responsible for coordinating the interoperation of roughly two thousand laboratory computers. Due to the decentralized purchasing, configuration, and management of each research group's computers, management of this network presents a remarkably difficult challenge.

As the Strata system came into operation, the laboratory networks came under heavy attack immediately following the Y2K rollover. The attacks appeared to begin with penetrations of vulnerable Linux machines, perhaps concentrating in groups with less vigilant patch maintenance. The core Strata machine, itself a Linux box, was among the machines under early attack, requiring a complete rebuild in mid-stream.

These hacked machines appeared to be used as bases for attacking other machine platforms. Soon, the routers of individual subnets began to crash, apparently under ping flood and denial of service attacks. On the eve of the SIGGRAPH paper deadline, the main building routers were themselves attacked and temporarily disabled, partitioning the building from the Internet.

During this period, many functions of Strata proved relevant to responding to the attacks, while others were added specifically for this purpose. For instance, the first two parameter wheels of Strata, machine selection by group and by operating system, made it easy to identify the Linux systems active in different laboratory groups. Parameter wheels selecting machines by subnet supported rapid response to subnets attacked with sniffers and compromised ssh shells. Parameter wheels performing SNMP router bandwidth monitoring assisted the detection of subnets under ping flood attacks.

Other database content and associated parameter wheel queries were added specifically to assist in identifying compromised machines, and tightening security measures on other platforms. For instance, SAINT, a descendant of the SATAN network vulnerability assessment tool, was run on selected subnets. Its results were imported into the SQL database, and boolean combinations with other parameter wheel tokens appeared promising

The Strata interface itself was in too experimental a stage to serve as a primary tool in the attack response. Nonetheless, the Strata system did play important roles in identifying several compromised machines. Moreover, the (ongoing) effort to respond to the attack has raised considerable interest in Strata's potential to help manage network complexities that are rapidly going beyond the reach of existing management tools.

3.4 Other Strata functionality

The Strata system includes several other functions which leverage its functionality off the strength of existing technologies. First, a mechanisms similar to the monitor slot functionality of mediaBlocks [10] is used to allow GUI access to the content of parameter wheel tokens. This allows new SQL queries to be bound to parameter wheels using GUI drag-and-drop. More broadly, this "gateway" allows Strata to be combined with mainstream network management platforms such as HP OpenView. This substantially increases the breadth of data, functionality, and real-world installed base that Strata can hope to leverage.

Secondly, as we will discuss further in the implementation section, Strata's layer visualizations were written in Java specifically to support their integration with Web-based access. For instance, in our laboratory context, while only a single Strata interface may initially be accessible for network management use, staff from many groups throughout the lab can remotely access and update Strata's databases through the Web. Here, the Strata and Web interfaces well-complement each other, each increasing the quality and quantity of information accessible by the other.

A related issue concerns how Strata's databases become populated in the first place. As discussed in the next section, we have implemented a number of scripts for automatically identifying – or in many cases, heuristically estimating – the identity of many kinds of information, such as the platform type and physical location of individual computers. However, such automation sometimes produces inaccurate data, and other times cannot derive the desired information without human assistance.

We have approached this issue in several ways. First, in implementing the GUI gateway and web-based access mechanisms described above, we have provided traditional pathways for information to be manually brought into the system.

Secondly, by adopting SQL as our core mechanism for managing Strata's internal information, we both increase the ease with which external data can be imported into our system, and with which Strata can access existing SQL databases. For instance, the trouble ticketing system used by our network managers, RT, centers around a SQL database, which considerably eases our interfacing efforts.

Thirdly – and of highest relevance from an interface perspective – we have integrated stylus-based manipulation into Strata's layer interface. This allows both pointer-based selection and manipulation (especially, movement) of spatial information.

Our belief is that this support will considerably ease the task of entering the spatial positions of new network and building resources, and modifying the positions of existing resources. Hopefully, Strata's substantial improvements in the ease with which this spatial information may be both expressed and perceived will increase user incentives for actively maintaining this information.

4 SYSTEM IMPLEMENTATION

The implementation of Strata is fairly evenly divided between hardware and software aspects. These are illustrated in Figure 5.

Strata's physical implementation includes the construction of both representational and mechanically structural physical elements; the design and integration of custom electronics; and the physical and logical integration of commercial devices.

On the software side, Strata efforts were divided between the design of its data acquisition and database functions, and those of its TUI, GUI, and Web-based user interfaces.

4.1 Physical implementation

Strata's tangible interface is dominated by two physical structures, each controlled by a discrete, physically representational element: the tiered system of physical layers, and the parameter wheel control panel.

The first of these is Strata's layer interface. Partially described in section 3.1, the layer interface is physically structured by a moderately-sized aluminum frame, custom machined and assembled for this use. This frame supports three moderately heavy flat panel displays, which at full horizontal extension of the drawer slides, extend without vertical support for more than 80cm. Especially given stylus-based interaction with these panels, structural stability was a concern and focus of design.

This frame supports three vertically-tiered flat panel displays. The top layer is a Wacom PL400 integrated LCD display and digitizer tablet. The lower two layers are conventional Mitsubishi LCDs. Each tablet is 38cm diagonal, and 1024x768 pixels in resolution. Three Wacom digitizer/displays were purchased and intended for integration, but production shortages have caused shipping delays of several months.

These three flat panel displays are mounted to commercial drawer slides with custom laser-cut acrylic jigs. The Mitsubishi panels are driven with a four-headed Appian Jeronimo graphics card, hosted on a Linux computer. The Wacom display requires an LVDS display, and is hosted on a dedicated Win95 computer.

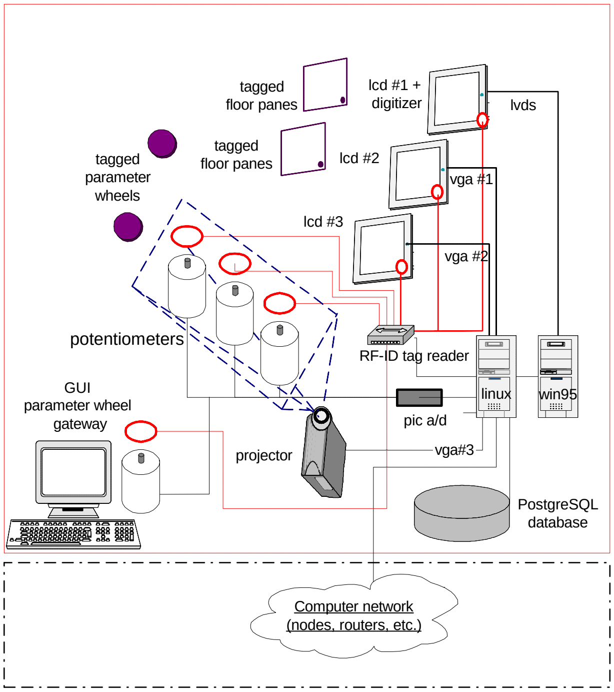

Figure 8: Diagram of Strata system implementation

The floor panes and parameter wheels were both constructed from laser-cut and -etched acrylic. They were embedded with RF-ID tags, using Philips Hitag2 protocol chips, and encased with coils in custom 2.7cm-diameter plastic disks by Cross Technology. These tags were sensed with a Micro RWD chip module, modified with custom electronics by *[colleague]* to multiplex the tagreader across four to eight sensor coils. We used these these tag readers in a four-coil configuration, resulting in per-coil update rates of roughly 3Hz.

The parameter wheel control panel was constructed of a vellum diffuser layer, sandwitched between two layers of transparent acrylic. An 800x600-resolution inFocus projector mounted on a custom laser-cut acrylic jig was used to back project computer graphics onto this surface.

The parameter wheel receptacles were made of a custom-designed laser cut enclosure. Our original parameter wheel receptacle design integrated an actuating motor/encoder assembly, to provide the user with computationally-actuated detents associated with each token parameter. This force-feedback actuation was seen as important, as it facilitated eyes-free parameter wheel manipulation when users' visual attention is focused on the Strata display layers.

However, due to time constraints, our current parameter wheel receptacle integrates a potentiometer assembly, which mechanically couples to parameter wheels. We augment this potentiometer interaction both graphically (through the back-projected control panel surface) and audibly, with audio providing some of the eyes-free cues originally provided with active force feedback. The three potentiometers are monitored with a PIC microcontroller circuit, and transmitted to the Linux machine via RS232.

We should note that in parallel, MacLean, Snibbe, and Levin have developed a similar force-feedback approach for token interaction that they call "tag handles." They offer a number of interesting design reflections in [6], which have strong relevance to Strata's evolving design.

4.2 Software implementation

Strata's software implementation is largely divided between two elements: automated data acquisition and database functions, and the code underlying its TUI and alternate user interfaces.

Roughly a dozen different data sources were parsed and cross-correlated for integration into Strata. Several different classes of data sources were used. A regularly-updated building phone list was parsed to acquire mappings between people names, login names, and office numbers. A database of internal mailing lists (some 2400 of them) was parsed and cross-correlated, yielding many kinds of user roles and affiliations such as student/staff, research group, etc.

Flat-text logs of programs like the MRTG router monitor were parsed to yield detailed time-logs of router activity. Custom SNMP monitoring jigs were also written with the Scotty Tcl extension. More powerful and aggressive tools such as NMAP and SAINT were eventually used to assess machine operating system type and security vulnerabilities. However, we have been very cautious in our use of NMAP, SAINT, and kindred programs, because of their unusually aggressive methods, and possible misinterpretations of intention by fellow network users.

One of our most interesting custom scripts processed the "last" login logs of the most heavily-trafficed laboratory computer, widely used for e-mail and online messaging. We used this data to heuristically estimate the physical location of individual machines. This was based on reverse-indexing the originating machines of users logins, and infer locations based on the histograms of login activity from these originating machines. This information was cross-correlated with user's office numbers and research group affiliations to yield machine location estimates which, while not always accurate, sometimes yielded more useful information than actual physical location.

4.2.1 SQL database integration

These processed data collections were first maintained in flat text files. A breakthrough came with the migration to the PostgreSQL relational database (RDB). Compared with flat files, the SQL RDB offered far more efficient, reliable, structured access to our increasingly complex data repositories. More importantly, SQL supported complex queries, standardized access grammars, and integration into complementary software systems such as the AOLserver web server.

One interesting feature of our implementation relates to the association of SQL queries with parameter wheels. Here, we store SQL commands and supporting data records directly into the SQL database itself, with variable parameters represented with '%s'-style substitutions.

This way, to interpret the results of a parameter wheel, we use SQL to resolve the tag ID into a SQL table containing a "live" SQL query, and then interpret and evaluate this SQL query at the time of visualization. We have found this approach to be far more flexible, powerful, and open-ended than the more ad-hoc approaches used in most TUI research systems to date.

4.2.2 User interface software

Strata's graphics software was written in Java. We made this decision for several reasons. First, we wanted physically separate users to be able to share the same view of the data inside the system. Second, the system's use of four graphical display devices made for substantial computer hardware requirements. Using code which could run on several platforms gave us more flexibility in structuring this hardware.

Both a Java applet and a Java application connected to our display server, which managed the communication between the display code and the rest of our system. We implemented the display server in Python so that it could rapidly evolve during the process of our resarch.

Our initial Java floor visualization applet only accepted simple commands from the server. While this was easy to implement, it meant that our displays performed poorly at times because large numbers of commands had to be sent over the network to perform operations that were conceptually very simple. To address this issue, we put more intelligence into the display code, giving it the ability to load and store display states by name. This ability to load and store statesalso maps cleanly to the SQL queries performed by our system. Typically, query results to be displayed are expressed based upon how the display should differ from a state already stored inside of the Java program.

5 RELATED WORK

We have mentioned the relation of Strata to a number of projects earlier in the paper, including mediaBlocks [10] and Urp [12], LogJam [1] and ToonTown [8], the metaDESK [11], tag handles [6], physical pixels [3], and distributed visualizations [13,14].

Beyond these, several other related works are relevant to our discussion. In the abstract, the Strata project is related to the mimic boards and map boards once common in electrical utilities and telecome centers, before becoming "obsoleted" by video walls. The greaseboards of aircraft carriers and perhaps other locales also are of relevance, especially to Strata's early designs in translucent layered acrylic.

Among research systems, the DispLayers use of multiple layered translucent graphics display surfaces, used in an interpersonal communications context, is quite relevant [5]. In the building visualization domain, Robertson, Card, and MacKinlay present one such 3D graphical system within a larger suite of information visualization efforts [7].

In addition to the information visualization sub-field, Strata has connections to several threads of augmented reality research. For instance, work by Feiner et al. uses head-mounted displays to render architectural walls "transparent," allowing users to see views of underlying physical infrastructure, etc.

The Financial Viewpoints research of [9] was partially inspirational to Strata. In particular, the layered transparent 3D graphical representations of this project were first "mocked up" with a layered translucent acrylic structure, providing us with partial inspiration for our original acrylic building models.

6 DISCUSSION

Several high-level considerations recurred repeatedly throughout the Strata design process. These include notions of "token and reference frame" systems, and issues of balance and appropriateness between physical and graphical representation.

Strata's early development grew in part from a study of board, card, and tile games, considered as systems of tokens and reference frames. Board games in particular invoke complex dependencies between physically manipulable tokens and generally static reference frames. Tokens and boards also take on a broad range of representational forms, from abstract marbles and checker boards, to more representational chess pieces and monopoly boards. Another spectrum lies between the hiding and revealing of information latent in card vs. board games.

The symbolic role of game pieces was referenced in the introduction. In considering possible analogies between these games and tangible interfaces, we found that a fairly wide variety of TUI "tokens" have been explored, but that development of provocative TUI "reference frames" has been more limited.

A number of tangible interfaces utilize a planar graphical surface as "reference frame" (e.g., the metaDESK [11] and Urp [12]). However, the marble answering machine [2], mediaBlocks [10], LogJam [1], ToonTown [8], and most recently tag handles [6] are to our knowledge among the only published systems to develop more complex, physical dependencies between tokens and reference frames.

Our original intention was to aggressively explore interactions between token and reference frames. In our first Strata implementations, network nodes such as routers were physically embodied as discrete tokens, which attached to the laboratory floor "reference frame" with peg-hole like structures. Here, our early implementations shared conceptual ground with Heaton et al.'s notion of "physical pixels."

However, after further discussions with our network management staff, they appeared to have strong interest in interacting with a great many network nodes – perhaps up to and including the current 2000 in-building computers. Where physically representing 10-20 routers had seemed interesting and compelling, representing 2000 nodes seemed to strongly suggest more malleable graphical representations.

Here, our leaning was towards projective technologies, but after building several prototypes, we fell back to the present use of flat panel technology, in no small part due to its implementational convenience. Nonetheless, we have maintained the use of projection with our parameter wheel interface, due to projection's affordance for "flowing" graphics "around" active mechanical structures. In time, we anticipate technologies like electronic ink or cholesteric displays will offer yet more practical technology.

Through this evolution, our current Strata implementation realizes a relatively conservative implementation of physically representational reference frames. However, we strongly believe the study of physical reference frames holds strong potential for tangible interfaces.

7 CONCLUSION

We have presented Strata, a tangible user interface supporting network and facilities management. Strata gives physical embodiment to layered information structures, in the process suggesting tangible interfaces specific to other increasingly more complex information structures. Strata also demonstrates methods by which tangible interfaces can provide rich access and manipulation of relational database content, extending its applicability to the broad span of commercial database-centric work.

Perhaps most provactively, Strata demonstrates a new class of computational interfaces whose structure is specific not only to a specific application, but also an embodiment of specific complex systems, whether in the semiconductor fab plant, the pharmaceutical lab, and beyond. We believe this work may point the way to new classes of tools which may have significant impact throughout the sciences, industry, and beyond.

8 ACKNOWLEDGEMENTS

Ommited for blind review.

9 REFERENCES

[1] Cohen, J., Withgott, M., Piernot, P. (1999). "Logjam: A Tangible Multi-Person Interface for Video Logging." In *Proc. of CHI'99*.

[2] Crampton Smith, G. (1995). The Hand That Rocks the Cradle. *I.D.*, May/June 1995, pp. 60-65.

[3] Heaton, K., Poor, R., and Wheeler, A. (1999). Nami. In *Conference Abstracts and Applications of SIGGRAPH'99, Emerging Technologies*, p.214.

[4] Ishii, H., and Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. In *Proc. of CHI'97*, pp. 234-241.

[5] Kobayashi, M. and Ishii, H. (1994). DispLayers: Multi-Layer Display Technique to Enhance Selective Looking of Over-

laid Images. Poster from *CHI'94 Conference*, April, 1994, Boston MA.

[6] MacLean, K., Snibbe, S., and Levin, G. (2000). Tagged Handles: Merging Discrete and Continuous Manual Control. To appear in *Proc. of CHI'00*.

[7] Robertson, G., Card, S., and Mackinlay, J. (1993). Information visualization using 3D interactive animation. *Communications of the ACM,* v36n4 (Apr. 1993), pp. 56-71.

[8] Singer, A., Hindus, D., Stifelman, L., and White, S. (1999). "Tangible Progress: Less is More in Somewire Audio Spaces." In *Proc. of CHI'99*.

[9] Strausfeld, L. Financial Viewpoints: using point-of-view to enable understanding of information. In *CHI'95 Companion*, pp. 208-209.

[10] Ullmer, B., Ishii, H., and Glas, D. (1998). media-Blocks: Physical Containers, Transports, and Controls for Online Media. In *Computer Graphics Proceedings* (SIGGRAPH'98), 1998, pp. 379-386.

[11] Ullmer, B., and Ishii, H. (1997). The metaDESK: Models and Prototypes for Tangible User Interfaces. In *Proc. of UIST'97*, pp. 223-232.

[12] Underkoffler, J., Ullmer, B., and Ishii, H. (1999). "Emancipaed Pixels: Real-World Graphics in the Luminous Room." In *Computer Graphics Proceedings* (SIGGRAPH'99), 1999, pp. 385-392.

[13] Yarin, P., and Ishii, H. (1999). TouchCounters: Designing Interac-tive Electronic Labels for Physical Containers. In *Proc. of CHI'99*, pp. 362-368.

[14] Yarin, P. (1999). Towards the Distributed Visualization of Usage History. MS thesis, MIT Media Lab, 1999.