HyperGrid – Accessing Complex Information Spaces

Hans-Christian Jetter, Jens Gerken, Werner König, Christian Grün & Harald Reiterer

Department of Computer & Information Science, Workgroup HCI, University of Konstanz, 78457 Konstanz, Germany
Email:*{Hans-Christian.Jetter, Jens.Gerken, Werner.Koenig, Christian.Gruen, Harald.Reiterer}@uni-konstanz.de*
URL: *http://hci.uni-konstanz.de*

In this paper, we describe a new interaction and visualization concept for non-expert users to easily access complex heterogeneous information spaces. The HyperGrid combines well-known table visualizations with zoomable user interface concepts to provide a two-dimensional grid as an access tool to multi-dimensional and relational data. It allows the presentation of attributes of multiple data types, various modalities and various levels of detail in a compact and consistent visual structure. Our concept is illustrated by a use-case based on our experiences and evaluations from preceding projects in the field of visual information seeking systems.

Keywords: HyperGrid, visualization, zoomable user interfaces, interaction techniques, heterogeneous information space, multi-dimensional data, information retrieval, evaluation.

1 Motivation

Many everyday tasks in our life are centred around decision-making based on accessing, sorting, comparing and relating multi-dimensional data. Imagine a salesman has to decide for one of several suggested flight connections at his local travel agent. To choose a connection which suits his requirements he needs to focus on multiple dependent variables simultaneously (e.g. travel time, number of necessary changes, ticket price). A highly efficient way to perform such

comparisons is collecting and structuring information in a table with the individual candidates in rows and the individual properties (or attributes) in columns. The orientation and sorting direction of the table allow a quick insight into similarities and differences among the candidates and reasonably increase the efficiency of decision-making.

Using information technology requires a large number of decisions based on multi-dimensional data as well: Everyday we decide for or against documents or web sites found by search engines or we select media objects like images or songs from large catalogues. These decisions are normally based on the given metadata sets for each candidate or object. In most cases this metadata is textual (e.g. name, URL), numeric (e.g. file size, date), and in some cases visual (e.g. thumbnails). As data formats and display sizes are well-defined, a traditional table with the metadata for each candidate displayed in the columns of the candidate's row can be used for visualization. The possibility to arbitrarily rearrange rows, columns or sorting orders has made the table to be a simple but very popular visual structure – even for navigation tasks in hierarchical data. Accordingly tables are not only used in classical spreadsheet or business applications but also as interfaces for presenting, selecting and manipulating file system or email objects (e.g. in Microsoft Windows Explorer or Outlook). In our preceding projects INSYDER and INVISIP evaluations of table-based visualizations [Gerken 2004] have shown that tables are never inferior and for certain tasks superior to lists in effectiveness and efficiency. Furthermore we have recently concluded a summative usability study with a non-expert user group (n = 24) to compare the web interface of the search engine for the library of the University of Konstanz (KOALA) with our table-based information seeking system MedioVis. It revealed significant (p<0.01) superiority in efficiency, appeal [Hassenzahl et al. 2003] and SUS scores [Brooke 1996]. However, loosely structured lists seem to remain the dominant design for the presentation of search results, document collections or other content on the Web.

The handling of metadata and content will become even more demanding for non-expert users regarding the future of Web applications: The information spaces of future e-commerce sites will constantly grow in quantity, dimensionality and in the heterogeneity of the contained data. In future search engines thumbnails, Flash animations or photographs of products will function as "teasers" to attract visitors and customers. Video-on-demand systems will not only contain posters or textual metadata, but a collection of video trailers, interviews, and other background information about featured films to advertise for buying access rights. Digital libraries or museums will increasingly contain complete scans from valuable historical originals (e.g. scrolls, books, maps), thus providing access to digitized full-text as well as catalogue metadata.

Regarding the growing challenge of handling content of this complexity, it is doubtful whether traditional list or table concepts can succeed on the long-run. Therefore it has been our focus during the development of the HyperGrid to create a strongly user-oriented interface to complex heterogeneous metadata and content at different levels of detail whilst remaining in a clearly structured and oriented visual environment. In the following we will introduce the key concepts of the HyperGrid and illustrate them in a use-case-scenario.

2 The HyperGrid Concept

The name "HyperGrid" is derived from the two key concepts on which our design is based on: "Hyper" stands for the ability to present interlinked hypertext or hypermedia content with different modalities, "Grid" stands for the two-dimensional orthogonal structure which is used for the interactive visualization. Although many aspects of the HyperGrid might remind of traditional tables at a first glance, the semantic structure of the grid and the role of the contained data cells differ fundamentally as we will outline in the following.

2.1 Cells, Columns and Rows in the HyperGrid

In our use-case and evaluations we used the HyperGrid to access a sub-set of the Internet Movie Database IMDb which is freely accessible under www.imdb.com. This online relational database is constantly growing and contains about 6.3 millions individual film or television credits and offers information ranging from the full cast of a film over the actors biographies and photos to user comments and reviews. Accordingly a film can have a remarkable number of attributes or metadata to describe it: the title of the film, actors, directors, producers, sound technicians, year of filming or length in minutes, long texts such as the plot or reviews, film posters or even video trailers.

Problems with table-based visualizations arise when the amount of attributes respectively columns exceeds the number that can be displayed simultaneously. The user is forced to either navigate horizontally through multiple pages or to reconfigure the selection of visible columns to reflect his current need of information. Problems even intensify when the information space to access is represented in a relational data model with great variances in the information density. Joining all tables and relations into one large table with numerous columns to "flatten" the information space will scatter data and leave gaps with no values assigned in between. For navigating such a large and sparsely filled table the user has to continuously keep an eye on his current position and orientation within the table and constantly recall his individual need of information. There is an immanent danger of getting lost in "table-space", a phenomena which is commonly known to Microsoft Excel-users working with large sheets. Furthermore special attributes of little importance for the average user can gain equally prominent positions within the table as those essential for the majority.

Like in traditional tables the cells of one row of the HyperGrid all belong to one individual data object or candidate. However, the columns are not assigned to single attributes but to groups of attributes with semantic proximity. Each of these groups or clusters of attributes represent one aspect of interest. The introduction of these higher-ranking aspects of interest is necessary to pre-process the amount of information for the user, who otherwise would face large unusable tables with dozens of columns. The assignment of attributes to columns is discussed in detail in chapter 2.3.

For our use-case we assigned the individual films to the data rows. Each film is characterised by three aspects of interest, which were assigned to the first three columns of the grid: "Film" representing information about the film itself, "Content" representing information about the plot or genre and "People Involved" containing directors, authors, cast, producers, etc. Figure 1 shows a screenshot of the fully functional HyperGrid from our test environment, showing those films from the IMDb which were retrieved by a simple keyword search.

Figure 1 – HyperGrid with film data from the IMDb.

As multiple attributes appear in one column the HyperGrid cells are not only static containers for one value. We regard every cell as one possible starting point for the exploration of the information space with the row defining the object of interest (the film) and the column defining the aspect of interest (e.g. "People Involved"). The third parameter controlled by the user is the degree of interest (DOI) [Rao & Card 1994].

In the HyperGrid the DOI is directly mapped to the level of detail and the size of visible information. Increasing the DOI leads to a growth of the cell size and more and more detailed attributes are fetched from the depth of information space and are presented on the HyperGrid's surface. Reducing the DOI leads to a reduction of the cell size and thus to a reduction of visible attributes. Therefore controlling the DOI allows smooth changes between abstract metadata, rich multimedia metadata, or full text in the cell (see figure 2). We refer to this mechanism as zooming.

Our interpretation of the HyperGrid cell as dynamic window is taken a step forward by including hypertext or hypermedia data. As mentioned above the IMDb

contains not only films but also persons as data objects. All entities of the database (e.g. actor, film etc) are carrying 1-to-n or n-to-n relationships among themselves. A film data set contains a list of actors; an actor data set contains a list of films.

The traditional way to provide such relational data on the web is to offer individual web pages for each entity. However, changing between them leads to a loss of the visual context and a change of layout or even modalities.
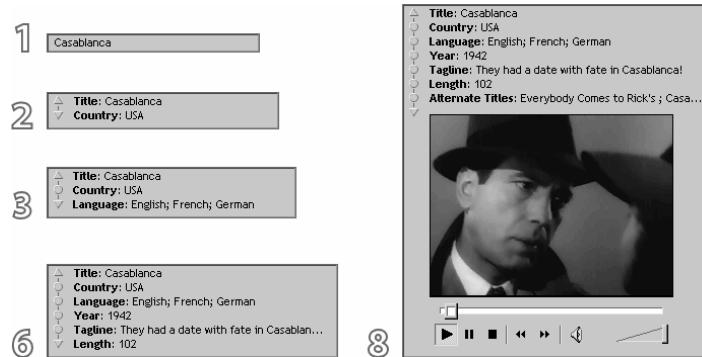

Figure 2 – Levels of detail with increasing DOI.

Figure 3 – Overlaying browser windows on top of the HyperGrid.

Since in our scenario films are the central object of interest there is no consistent way to integrate person-related biographies or filmographies directly into the rows. One solution would be to switch the whole interface from a film-related to a new

person-related grid, if requested by the user. However this would lead to a complete loss of the context from which the information was requested. For this reason we prefer the concept of transforming grid cells into small browser windows to allow the user excursions into other objects of interest. This concept combines the browsing of conventional hyperlink structures with table-based visualization: Every object from the information space which carries additional information – but is not the central object of interest – is represented via a classical hyperlink within the cells. For example, if a user has the need to get more information about an actor from a HyperGrid displaying films, he can click on the actor's name as hyperlink. Instead of switching to a new person-related table, a browser-like window is opened in front of the corresponding cell (see figure 3).

Unlike standard web browsers the browser window in the grid is integrated directly in the link's context. Neighbouring cells, selected objects or sort orders remain preserved. If more space for the window is necessary, its size can be gradually enlarged up to a size filling the complete grid area. Therefore we reduce the changes of modalities and the danger to get "lost in hyperspace".

Additionally the overlaying browser windows can be used to fully display attributes which are especially space-consuming like long texts, lists or video-trailers. Since some attributes need to be truncated in the cell because of space limitations, the user can open an overlaying window by clicking on the "…" or an arrow icon behind the shortened content (see figure 3, plot "Batman Forever").

2.2 Visual Information Seeking with the HyperGrid

In the following we will discuss the HyperGrid against the background of Shneiderman's "Visual Information Seeking Mantra" [Shneiderman 1996]. Through leaving out subtle distinctions Shneiderman identifies two diametrically opposed kinds of information need: "Known-item search" and "Browsing". In our understanding the first assumes that the user has a specific information need which he expresses for example by entering precise keywords in a search engine or navigating to specific nodes in an information hierarchy. Browsing-oriented access to information spaces assumes the contrary: The user has only a vague and less specific information need which he can hardly express in keywords or by systematic navigation. While browsing the information space the user gradually refines his information need by deciding for or against information he is confronted with. To support user goals reaching from known-item search to browsing, Shneiderman identifies seven tasks in his "task by data type taxonomy": Overview, zoom, filter, details-on-demand, relate, history, and extract. Based on this framework we will discuss how the HyperGrid succeeds in providing task-oriented functionality to support the user in reaching his individual goals.

2.2.1 Overview

Whether the HyperGrid displays a whole catalogue of thousands of items or just dozens of results from a search query is a question of the application domain. Our intention was to develop the HyperGrid as a generic visualization which works

independently from previous information retrieval. Figure 4 shows the initial state of the HyperGrid offering an overview of search results retrieved with "Martin Scorsese" specified as keyword. However the table content could also be an entire film collection catalogue without previous searching.
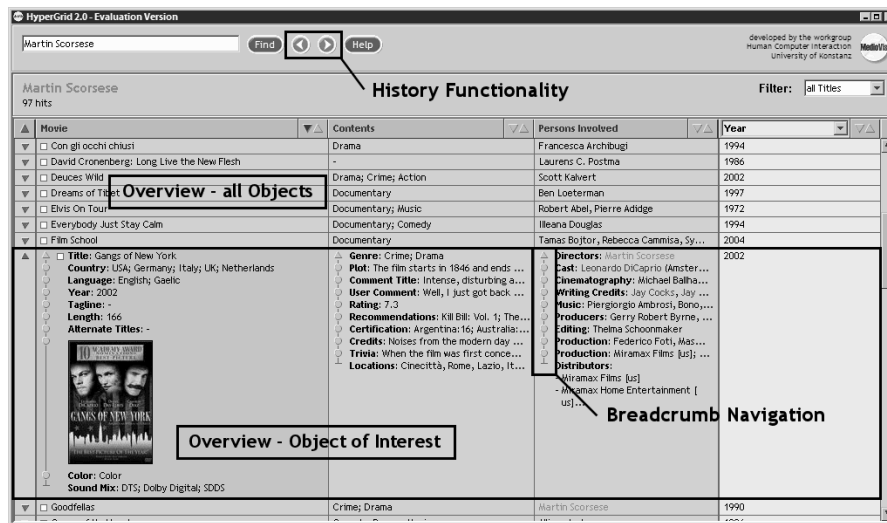

Figure 4 – HyperGrid Overview.

The HyperGrid does not offer a graphical overview based on scatter plots [Ahlberg & Shneiderman 1994] or distortion techniques from the TableLens [Rao & Card 1994]. This decision is based on experiences from our previous projects in the field of visual information seeking systems. There are only very few quantitative attributes in our use-case, which significantly reduces the benefit of mapping data to visual variables like position of a data point or the length of bars. Regarding our project VisMeB [Klein et al. 2003] we have observed that purely visual overview techniques often lead to a small conformity with user expectations. During our user tests these problems lead to a complete ignorance of the users towards the integrated TableLens concepts [Gerken 2004]. This observation led to the decision to reduce overview in the HyperGrid to the initial appearance of a table, without distortion effects or bar representations of quantitative data.

Although a table is a rather traditional visual structure it incorporates high conformity with user expectations with the advantages tables have compared to loosely structured result lists common in web-based applications. As a non-expert access tool the HyperGrid does not strain the user with unexpected visual structures in an initial state. However advanced techniques like scatter plots can be offered optionally outside the grid and can be coupled with the HyperGrid with brushing and linking techniques as realised in our MedioVis (see chapter 1) system.

Furthermore we provide an overview of a single object of interest. This function is realised via the small arrow icon in the very first column on the left. By clicking the user can expand a row to the maximum size and request an overview of all

attributes that are available for an object of interest at any time (see figure 4, film "Gangs of New York"). He can then choose the attribute of interest directly without having to navigate through the whole information space by zooming into cells.

2.2.2 Zoom

Figure 2 illustrates how zooming techniques allow increasing the level of detail within a cell while preserving the user's sense of position and context. Zooming interaction is provided by simple mouse clicks as suggested by Shneiderman [1996]. Unlike conventional zooming the HyperGrid allows selective zooming into one of multiple dimensions: By clicking into an individual cell the user expresses a higher DOI for an individual aspect of interest of an individual object.

Since a table structure does not allow completely independent cell sizes (vertical and horizontal size of a row or column is always defined by the cell with maximum size in this particular row or column) unintended growth of neighbouring cells during zooming has also to be addressed. Our concept is to compensate such an unintended growth of cell size by automatically filling the resulting space with information. If a user has chosen one aspect of interest and starts to zoom into one cell, he has implicitly expressed an interest into the whole object as well. Thus we carefully increase the level of detail in the neighbouring cells to enable the user to gradually get an overview of the whole information available and reveal other potentially interesting attributes.

2.2.3 Filter

Possible filter techniques are strongly dependent from the application domain and available data types. Trying to provide a generic concept, the standard HyperGrid contains only very simple filter functionality. In figure 4 a dropdown box is situated in the top right corner of the screen with three possible selections for the display mode: "all titles", "selected titles", and "zoomed titles". As proposed by Shneiderman [1996] this dropdown box allows the user to control the contents of the display and to filter out uninteresting objects. In a first step the user defines a sub-set of all interesting objects by selecting or zooming into them. In a second step a filtered display is activated by the user, selecting the desired display mode from the dropdown box.

Leaving the generic concept of the HyperGrid allows more powerful but domain-specific filter techniques: Apart from dynamic queries like in FilmFinder as proposed by Ahlberg and Shneiderman [1994] we suggest the table filter inspired by Microsoft Excel's auto filter function. The table filter reduces the displayed objects in the table to the sub-set which corresponds to the filter criteria specified in the top row of the table (see figure 5). For each column a value can be entered into the top row which defines the keywords that must be contained in the attributes of the corresponding aspect of interest. If the keyword is not found, the object is filtered out. Only those objects are displayed which fulfil all criteria in each column simultaneously (AND-relation). This way the amount of information

displayed can be reduced to the user's information need very efficiently without new search queries or database accesses.

Figure 5 – The table filter with one criterion for the "Content" column.

2.2.4 Details-on-demand

As illustrated in 2.1 the hyperlink concept and the overlaying browser windows allow easy access to details which cannot be integrated consistently into the table cells (see figure 3). Offering the functionality of a small browser application, each browser window can be used as a starting point of browsing-oriented excursions into the information space or connected information resources from the World Wide Web. Similar to the zooming in 2.2.2 the user's sense of position and context is preserved.

2.2.5 Relate

Based on the user's selection of one or several quantitative or categorical attributes as sorting criteria, table-based visualizations allow quick and systematic comparisons between several objects of interest. To further support this functionality, we included an additional user-adjustable column into the HyperGrid. This column is visually separated from the other columns and contains only one attribute which is selected by the user from the variety of attributes contained in the aspects of interest (see figure 6). As a result relating, comparing, and sorting values of an attribute that is hidden deep inside the information space can be achieved easily without navigating the aspects of interest at high levels of detail.

Figure 6 – The user adjustable column.

In a future version the hyperlinks described in 2.1 could also be used to analyse relationships among objects of interest by using them as input for query refinement or dynamic queries. Hyperlinks to objects of interest that are contained in the HyperGrid could also be used to jump to particular positions in the grid or to expand or select related rows.

2.2.6 History

History functionality is integrated into the HyperGrid on two levels: On a first level each interaction step that changed the face of the HyperGrid (e.g. zooming in or out, expanding or collapsing a row, opening a browser window) is stored in a browsing history. Users are offered an undo-function by clicking forward / backward buttons as known from web browsers (see figure 4). This very simple implementation of history is motivated by our observations from early user testing, where many users had the strong tendency to move the mouse pointer to the top left corner of the screen, obviously searching for a button to undo changes or returning to previous states of the HyperGrid. In some cases such functionality was explicitly requested by the users referring to popular interfaces like Microsoft Internet or Windows Explorer.

On a second level a visualization of the history of information growth is integrated into each cell. Since it is difficult to communicate the outcome of the zooming interaction step in advance, the information growth within a cell must be transparent for the user. Otherwise the user might get the feeling of losing the ability to control the DOI. For this reason the amount and kind of information added with zooming is illustrated with a vertical breadcrumb navigation within each cell (see figure 4). Similar to breadcrumb navigations used on the World Wide Web (e.g. to visualise the progress of the checkout process in online e-commerce shops), the user has the possibility to navigate back to an attribute by clicking on the breadcrumb and to retrace his path through information space.

2.2.7 Extract

As the HyperGrid shall be introduced as a generic visualization and interaction concept, it is not possible to discuss the extraction of data for later use in general terms. It is necessary to relate to specific application domains to illustrate the possibilities the HyperGrid can offer in future applications. One typical example is the application of the HyperGrid in digital libraries. Here the current selection of objects of interest from the grid could be saved to a portable medium, printed out or sent by e-mail like it is possible in MedioVis. Advanced systems can manage multiple collections per user or allow the exchange of collections among them for collaborative work.

2.3 The Attribute Space

To illustrate the concept how attributes can be assigned to aspects of interest (respectively columns) we have introduced the attribute space, which lays a simple mathematical foundation for structuring data in the HyperGrid and offers possibilities for future work concerning an optimised or user-adaptive information presentation and assignment.
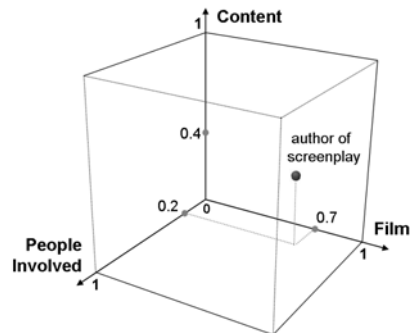
Figure 7 – An attribute in the attribute space.

According to our IMDb use case we want to assign each attribute of a film to one of the three aspects of interest mentioned above ("Film", "Content" and "People Involved"). In the attribute space every aspect of interest is equivalent to a dimension (see figure 7). Attributes are equivalent to points in attribute space, but are limited to positive Cartesian coordinates between 0.0 and 1.0. In our example the attribute space is therefore 3-dimensional and contains a cloud of points limited to a cube of edge length 1.0 in positive direction of Cartesian space.

The location of the attributes in the attribute space is defined by one or multiple experts by assigning a semantic distance to each attribute for every aspect of interest. Assigning 0.0 as location for an attribute on an axis means a very close relation to this aspect of interest (respectively a small distance to the origin). A 1.0 means there is practically no semantic proximity of this attribute to the aspect of interest. For example an attribute containing "author of screenplay" is very closely related to the 3rd dimension "Persons Involved", however it has also reasonable influence on the 2nd dimension "Content". Regarding the 1st dimension "Film" authors are normally less interesting than actors or directors. Accordingly an expert could assign the location (0.7, 0.4, 0.2) to the attribute "author of screenplay". Without any doubt this is a subjective assessment which might vary between experts with different mental models and needs of information. However this weighed assignment of attributes to aspects of interest allows various strategies how to project attribute space onto the HyperGrid columns or to cluster or join attributes of small semantic distance.

At this early stage we were already able to achieve reasonable results by merging the assessment of two members of our workgroup who were supported by a researcher in the field of media science. The merging was simply done by

calculating the average position for each attribute and assigning the attribute exclusively to the aspect of interest it was most related to.

In future more intelligent strategies and dynamic adaptation during usage could improve user satisfaction with data structuring in the HyperGrid. Using our DROID logging technology [Jetter 2003] we are able to collect information about the popularity of attributes and number and duration of visits. Like suggested in the representation matrix of Rüger et al. [1996] such information could be used to optimise attribute space on the basis of simple statistical mathematics, without complex user modelling or inferences on user-behaviour.

3. User Testing and Redesign

As mentioned before our intention behind the HyperGrid is to offer an intuitive interaction and visualization concept for non-expert users. Therefore development has been accompanied from the scratch by intensive use of inspection methods to verify the concepts at all stages of implementation. However user testing is essential to measure how far these efforts were successful. Since the maturity of our theoretical concepts and their implementation in our Java prototype has not yet reached a stage to measure effectiveness and efficiency and compare them with existing systems, we were focussed on identifying fundamental problems in our concept and therefore conducted a formative evaluation. Nevertheless our HyperGrid Java implementation was already fully functional.

3.1 Test Setting

We tested a group of five non-expert users with strong differences concerning computer literacy with the above mentioned IMDb use-case. The test was carried out in our usability lab with one test person per session. Our test-design was a mixture of several qualitative methods to increase the validity of the outcomes: observing user's behaviour and comments during an unguided exploration phase, "thinking aloud" user testing with realistic tasks, and a concluding semi-structured interview. Two questionnaires (Attrakdiff [Hassenzahl et al. 2003, www.attrakdiff.de] and Software Usability Scale (SUS) [Brooke 1996]) were included to measure additional quantitative data. Because of the small size of the test group complex statistical analysis was not regarded as reasonable.

3.2 Test Results

3.2.1 Usability Problems & Redesign Approaches

We could identify two central usability problems. First users felt that the content of a cell changed rather randomly during zooming and was hardly predictable. This problem occurred because of the layout and truncation algorithm which failed to provide a consistent layout within the cell while simultaneously guaranteeing a steady increase of cell size during zooming. As a consequence further effort must be put into the improvement of the algorithm. The second problem originated from

too many interaction possibilities that were offered in each single cell. Users could zoom in, zoom out, click on the breadcrumb arrows to zoom in/out entirely, click on a breadcrumb or attribute name to focus one attribute, and last but not least click on hyperlinks to see additional information on a person or an attribute in a browser window. Whereas this variety of options might be suitable for experienced users, it seems not adequate for non-experts.

Since our focus lies especially on ease-of-learning, we will now aim at reducing the amount of different interaction elements. Therefore we have to rethink the breadcrumb navigation. Its initial idea was to visually structure the information growth during zooming in order to improve comprehensibility for the user. However it remains questionable if it succeeded in this task. Furthermore the interaction possibilities were mainly redundant. We will therefore evaluate ways to replace the breadcrumb navigation completely which will reduce interaction possibilities within a cell to zooming and hyperlinks (e.g. person names for biographies or attribute names for full view of attributes).

3.2.2 Approved Concepts

Besides the above mentioned conceptual flaws the fundamental concepts of the HyperGrid were approved:

1. All users seemed to comprehend the mechanism of zooming into and out of cells and could clearly relate cell content to the table context.

2. Controlling the DOI by mouse clicks was no challenge: The affordance generated by changing the shape of the mouse pointer over the cell appeared to be sufficient.

3. The HyperLink concept to integrate additional information in separate browser windows seemed to fit the users' needs very well. Users comments varied from "nice to have" to "really cool" and users enjoyed the intuitive interaction.

4. The user adjustable column was considered as extremely helpful to compare several movies with one specific attribute.

5. The sorting functionality was discovered by all users thanks to larger and more meaningful sort buttons in the column header.

6. The filter functionality to show only selected or zoomed rows was understood and adopted. Especially in large result sets, our subjects made frequent use of it. Nevertheless we will integrate a superior filter concept directly within the table visualization as mentioned in 2.2.3.

Finally the two questionnaires Attrakdiff and SUS underlined the overall positive feedback: Although users ran into some usability problems, the questionnaire results suggest that users liked the HyperGrid in terms of functionality (SUS) and

hedonic quality (Attrakdiff). While the small sample size doesn't allow an in-depth interpretation of the exact results, they are listed below for the sake of completeness: The SUS scored with a 71 on the 0-100 scale, which we regard as an acceptable value for a system still under development. Average Attrakdiff scores (scale 1.0 to 7.0) lie at 4.9 for pragmatic quality, at 5.2 for hedonic quality (identity), at 5.4 for hedonic quality (stimulation), and at 5.3 for appeal. Hedonic quality and appeal seem therefore already very satisfying.

4. Related Work

The extension of traditional tables by using distortion techniques was introduced by the TableLens [Rao & Card 1994] and was further developed by Spenke et al. [1996] and Bederson et al. [2004]. TableLens and Bederson's DateLens are focussed on distortion and semantic zooming and are the foundation of the HyperGrid's concept of combining zoomable user interfaces [Raskin 2000] with grid structures, semantic zooming and browsing (see chapter 2.1, 2.2.1).

In our preceding projects INSYDER and INVISIP we have extended traditional tables to control the level of detail of displayed metadata for entire tables or single rows [Reiterer et al. 2000], [Mann 2002], [Göbel et al. 2002].

The assignment of multiple attributes to "aspects of interest" at different degrees of interest is based on the representation matrix introduced by Rüger et al. [1996] which lays the foundation for the attribute space (see 2.3).

The HyperGrid's novelty lies in merging concepts of TableLens and DateLens with a cell-oriented semantic zooming to gradually reveal complex heterogeneous metadata attributes and content of different modalities according to the "attribute space" model. Furthermore it allows browsing-oriented excursions into the information space or connected information resources from the World Wide Web.

5. Conclusion and Outlook

In the past various technologies have been introduced to access complex information spaces and support decision-making by visualising multi-dimensional data. For example Attribute Explorer [Tweedie et al. 1994], InfoZoom [Spenke et al. 1996], hyperbolic trees [Lamping & Rao 1994] or scatter plots [Ahlberg & Shneiderman 1994] can largely simplify decisions by exploiting the human visual processing to reduce the cognitive load of many tasks.

However these technologies are seldom found in standard applications or large e-commerce web sites to this day. It seems that the loosely structured list-based presentations on web-sites and the traditional table widget in desktop applications remain dominant because they have a small or irrelevant learning threshold and a great conformity with user expectations.

We are convinced that the usability of information systems could be largely improved by replacing poorly structured hypertext or list-based presentation of multi-dimensional data by the HyperGrid.

Based on the concept of the Granularity Table [Klein et al. 2002] that reduces cognitive overload by smooth transitions between focus and context and

integrating data of different modalities in one data row the HyperGrid takes a step forward: the user can express his individual degree of interest into every table cell by a zoom-like interaction. Furthermore the HyperGrid does not lack the ability to deal with non-quantitative, non-categorical or non-textual metadata (e.g. images, video and sound clips) and can handle hyperlinks between different data objects enhancing the application domain from flat data tables to relational databases. The HyperGrid's close relation to well-known table concepts improves the ease-of-learning by the great conformity with user expectations. The apparent structure, orientation and the possibility of comparing and relating data should sustain a superior ease-of-use.

We are convinced that a matured design of the HyperGrid ready for deployment opens new perspectives for information presentation on web sites and the design of information seeking systems. Introducing a standard component like the HyperGrid to access complex information spaces (regardless whether it is flight connections, Wikis, e-commerce or digital library catalogues) could improve efficiency and the consistency of web sites with user expectations.

Furthermore the HyperGrid could be extended to additionally offer user interface components. These components could be placed as interactive objects in the depth of information space like normal textual attributes, video trailers or graphics. This way it would be possible to use the grid structure to integrate text fields or sliders for user comments or ratings into digital libraries or to implement e-commerce processes (e.g. ordering a product or requesting support) directly into catalogues.

References

Ahlberg, C. & Shneiderman, B. [1994a], Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, in B. Adelson, S. Dumais & J. Olson (eds.), Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating Interdependence (CHI'94), ACM Press, pp.313–7.

Ahlberg, C. & Shneiderman, B. [1994b], Visual Information Seeking using the Filmfinder, in C. Plaisant (ed.), Conference Companion of the CHI'94 Conference on Human Factors in Computing Systems, ACM Press, p.433.

Bederson, B. B., Clamage, A. D., Czerwinski, M. P.& Robertson, G. R. [2004], DateLens: A Fisheye Calendar Interface for PDAs, ACM Transactions on Computer–Human Interaction 11(1), 90–119.

Brooke, J. [1996], SUS: A Quick and Dirty Usability Scale, in P. W. Jordan, B. Thomas, B. A. Weerdmeester & I. L. McClelland (eds.), Usability Evaluation in Industry, Taylor & Francis.

Gerken, J. [2004], Evaluation of a Metadatabrowser — Listview vs. Leveltable, Bachelor-Thesis, University of Konstanz.

Göbel, S., Haist, J., Reiterer, H. & Müller, F. [2002], INVISIP: Usage of Information Visualization Techniques to Access Geospatial Data Archives, in A. Hameurlain, R. Cicchetti & R. Traunmüller (eds.), Database and Expert Systems Applications: Proceedings of the 13th International Conference (DEXA 2002), Vol. 2453 of Lecture Notes in Computer Science, Springer-Verlag, pp.371–80.

Hassenzahl, M., Burmester, M. & Koller, F. [2003], AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität, in J. Ziegler & G. Szwillus (eds.), Mensch & Computer 2003: Interaktion in Bewegung, B G Teubner, pp.187–96.

Jetter, H.-C. [2003], Usability Evaluation im Rahmen von INVISIP, Bachelor-Thesis, University of Konstanz.

Klein, P., Müller, F., Reiterer, H. & Eibl, M. [2002], Visual Information Retrieval with the SuperTable + Scatterplot, in E. Banissi (ed.), Proceedings of the Sixth International Conference on Information Visualization (IV'02), IEEE Computer Society Press, pp.70–5.

Klein, P., Müller, F., Reiterer, H. & Limbach, T. [2003], Metadata Visualization with VisMeB, in E. Banissi, K. Borner, C. Chen, G. Clapworthy, C. Maple, A. Lobben, C. Moore, J. Roberts, A. Ursyn & J. Zhang et al.(eds.), Proceedings of the Seventh International Conference on Information Visualization (IV'03), IEEE Computer Society Press, pp.600–5.

Lamping, J. & Rao, R. [1994], Laying Out and Visualizing Large Trees using a Hyperbolic Space, in P. Szekely (ed.), Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology, UIST'94, ACM Press, pp.13–4.

Mann, T. [2002], Visualization of Search Results from the WWW, PhD thesis, University of Konstanz.

Rao, R. & Card, S. K. [1994], The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus and Context Visualization for Tabular Information, in B. Adelson, S. Dumais & J. Olson (eds.), Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating Interdependence (CHI'94), ACM Press, pp.318–22.

Raskin, J. [2000], The Humane Interface: New Directions for Designing Interactive Systems, Addison–Wesley.

Reiterer, H., Mußler, G., Mann, T. & Handschuh, S. [2000], INSYDER — An Information Assistant for Business Intelligence, in N. Belkin, P. Ingwersen & M.-K. Leong (eds.), Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00), ACM Press, pp.112–9.

Rüger, M., Preim, B. & Ritter, A. [1996], Zoom Navigation: Exploring Large Information and Application Spaces, in T. Catarci, M. F. Costabile, S. Levialdi & G. Santucci (eds.), Proceedings of the Conference on Advanced Visual Interface (AVI'96), ACM Press, pp.40–8.

Shneiderman, B. [1996], The Eyes Have It: A Task by Data-type Taxonomy for Information Visualizations, in Proceedings of the 1996 IEEE Symposium on Visual Languages (VL'96), IEEE Computer Society Press, pp.336–43.

Spenke, M., Beilken, C. & Berlage, T. [1996], FOCUS: The Interactive Table for Product Comparison and Selection, in D. Kurlander, M. Brown & R. Rao (eds.), Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology, UIST'96, ACMPress, pp.41–50.

Tweedie, L. A., Spence, B.,Williams, D.& Bhogal, R. [1994], The Attribute Explorer. Video presentation at CHI'94.