Context and Adaptivity-Driven Visualization Method Selection

Maria Golemati[1], Costas Vassilakis[2], Akrivi Katifori[1], George Lepouras[2], Constantin Halatsis[1]
[1] Department of Informatics and Telecommunications, University of Athens, Greece
[2] Department of Computer Science and Technology, University of Peloponnese, Greece

ABSTRACT

Novel and intelligent visualization methods are being developed in order to accommodate user searching and browsing tasks, including new and advanced functionalities. Besides, research in the field of user modeling is progressing in order to personalize these visualization systems, according to its users' individual profiles. However, employing a single visualization system, may not suit best any information seeking activity. In this paper we present a visualization environment, which is based on a visualization library, i.e. is a set of visualization methods, from which the most appropriate one is selected for presenting information to the user. This selection is performed combining information extracted from the context of the user, the system configuration and the data collection. A set of rules inputs such information and assigns a score to all candidate visualization methods. The presented environment additionally monitors user behavior and preferences to adapt the visualization method selection criteria.

Keywords: Person/machine interaction, Information Presentation, Information search and retrieval, Knowledge-Based Systems

Introduction

New visualization systems are continually equipped with advanced features in order to enhance search and browsing activities. However, regardless of a thorough visualization design, systems remain unable to satisfy any possible need and task. This is due to not only the huge volumes of information which exist in digital format and the diversity in digital collections' parameters, but also because users who rely on electronic media in order to forage the information they need, often come up with specific and complicated demands. As this new era in information approach is getting shaped, a number of extra factors emerge.

To effectively achieve an information retrieval goal, any individual user's characteristics play a decisive role as they present different behavior when solving different tasks or even the same task under different circumstances. Thus, recording these characteristics a system could be evaluated as suitable or not for a specific user or user group. On the other hand, the particularities of a searching task, as well as the corpus which hosts the possible results, provide key information for the effectiveness of a specific system in the solution of a specific task. Consequently, any individual visualization system is never enough for any possible need and task.

The development of user-adaptive systems is a promising approach to address this problem, as these systems are designed to be customized to the needs and desires of their specific users. Building and then exploiting user models, user-adaptive systems incorporate dynamic processes which allow humans to define their function according to the surrounding situation.

In this paper, the user characteristics, the data collection particularities and the system capabilities are matched with the visualization method properties in a context-based adaptive visualization environment to be used in the Historical Archive of the University of Athens, in order to support information seeking tasks. The presented work introduces new techniques for supporting the adaptation and personalization issues in the design and development of Intelligent User Interfaces, mainly by adapting services to user preferences and device characteristics of the user (display and input devices available), while system constraints and resource availability (memory size and processor speed) are also taken into account.

In the next section of this paper, background issues and related work in the field of user modeling and user adaptive systems is surveyed. In the following sections, we analyze the notion of context modeling in our system and describe the process of visualization method selection, which is also exemplified through an hypothetical user session with the proposed system. In the last two sections, future trends are discussed and conclusions are drawn.

Background

The problem of context management constitutes a new approach to the design of context-aware systems. (Zimmermann A., Specht M. & Lorenz A., 2005) refers to this problem combining personalization and contextualization. It defines that an *adaptive system (contextualized and personalized or both) follows an adaptation strategy (e.g. pacing or leading) to achieve an adaptation goal (e.g. intuitive information access or easy use of a service). To achieve an adaptation goal, it considers relevant information about the user and the context and adapts relevant system components on the basis of this information*".

(Domik G. O. & Gutkauf B, 1994) claims that a visualization system needs to adapt to desires, abilities and disabilities of the user, interpretation aim, resources (hardware, software) available, and the form and content of the data to be visualized. It distinguishes four different models: user model, problem domain/task model, resource model and data model and gives the design of computer tests and games to test user abilities (color perception, colour memory, colour ranking, mental rotation and motor coordination). (Fischer G., Lemke A., Mastaglio T., & Morch A., 1991) suggests the following three kinds of user modeling:

- the explicit modeling, which involves asking the user straightforward questions. Such kind of information is usually collected in the beginning of the user's interaction with the system.
- the implicit modeling, according to which the system extracts information from the user's work and interaction with the system. For example, recording the keys pressed and functions used, or the choices a user makes.

- special tasks to solve, where the user is submitted in solving special predefined tasks, designed for the purpose of extracting certain abilities of a user.

Each type of model further individualizes and enriches the information of the previous one(s).

The IVEE (Ahlberg, C., & Wistrand, E., 1995) is a visualization system, which supports multiple views of the data collection together with a functionality to format a query in a dynamic way. According to the authors of the system, a successful visualization environment depends on a whole set of visualizations appropriate for various tasks and data types, as it can be customized in a variation of existing conditions. In IVEE this notion is applied, providing the user with a variety of visualizations and features to customize to different preferences, abilities and needs. The system is only implemented in the context of movie searching and does not support document properties such as hierarchical structure, hypertext structure etc.

In the Periscope system (Wiza W., K. Walczak & W. Cellary, 2004) a holistic, an analytical, a hybrid as well as a specialized interface model have been implemented both in 2 and 3 dimensions to give the user the opportunity to select a specific presentation method to focus on certain properties of the results obtained. The system allows the user to assign search result attributes to visualization dimensions and therefore modify the method of visualization to highlight important features of the search result. Furthermore, the system provides the possibility to make comparisons between results from two or more different queries in a single 3D scene.

The problem discussed in this work is a specific instance of the generic problem of expressing and evaluating user preferences discussed, which has been discussed, among others, in (Agrawal, R. & Wimmers, E. L., 2000) and (Seunghwa, L. & Eunseok, L., 2007). In (Agrawal, R. & Wimmers, E. L., 2000), a generic scheme is proposed, which allows autonomy and combination of various preferences. The personal preference model used in this work follows these basic principles accommodating additionally the context parameters which held when some user preference was expressed. Under this enhanced scheme, when some preference is considered the context parameters recorded in the preference are compared to the ones currently effective and the result of this comparison indicates how strongly this preference should be taken into account.

Based on the above research and taking into consideration the added value of the user and other feature modeling we suggest an adaptive visualization environment which adapts to specific users, tasks and environments. The result is a novel context-sensitive information space, which adjusts its appearance and functionality to best serve the user in any given situation.

Context Modeling

Modeling the context of the Historical Archive consists not only of the user characteristics, preferences and needs, but also of the platform available to perform the task and moreover the properties of the document collection to be retrieved. Consequently, the concept of context modeling includes each of the following cases:

Explicit user context: this is the initial user information. For example, the gender, the age, the profession, the educational level, the cognitive abilities, the user's experience in using computers and so forth, may constitute the explicit user context. Such information can be extracted through interviewing the user the first time s/he accesses the Historical Archive and it is used to initially populate the specific user's *preferences database*. For example, for users with little computer experience a low preference score for visualization methods with complicated controls is recorded, whereas for users with elevated color perception, high preference scores towards methods using color coding are registered. Demographic data (e.g. gender and age) and personal information (e.g. profession) are used to classify the user into a *user stereotype*, which is also associated with a set of preferences; this classification eases the initial preference database population.

These preferences are extracted from the user's preferences database the next time the same user visits again the Historical Archive, and are taken into consideration in the process of selecting the set of candidate visualization methods for the task at hand. Note that these initially set preferences may be later modified if the users' behavior in the system suggests that the recorded preferences do not apply as registered.

Implicit user context: this is additional information extracted while the user is working with a single visualization method. For example, his/her preferences and/or likes/dislikes of the visualization methods, his/her difficulties in understanding a method in finding the information needed etc, constitute the implicit user context. This information is registered to enhance the user profile with additional data which will also be considered in any future selection of the appropriate visualization method for the specific user.

System context. This concerns information relative to the software/hardware available to perform the visualization. For example, the existence or not of VR equipment, the memory size and processor power are elements of the system context. The system context is exploited by the adaptation mechanism to determine the most appropriate visualization method to employ. For example, a 3D visualization method can be more efficiently displayed when 3D monitors and/or other VR equipment is available. Note that the absence of such hardware may not preclude the use of 3D visualization methods, and, inversely, their presence does not imply that only 3D methods will be used; system context is co-evaluated along with other parameters to finally select the most prominent visualization method.

Document collection context. This concerns information relative to the portion of corpus of the Historical Archive that will be visualized in a given situation. Such a corpus has many particularities. For example, it may contain documents in various formats (text or scanned images, while for some documents only their *metadata* may be available); The documents may be related or not according to some criterion; selected documents may be classified under a taxonomy or they may have been retrieved using a query; and so forth. The most special case of the Historical Archive data collection is the minutes from the various University meetings, since a single document of this class may span across different thematic categories, affect multiple departments, reference or be referenced by multiple other documents etc. Consequently, these documents require a different visualization method, depending both on the data characteristics and on the user's needs as well.

The above-mentioned contexts provide valuable knowledge to the process of the selection of the most appropriate visualization method to display the information. This knowledge is processed using a set of rules, assigning to each method a score – effectively a value indicating its perceived usefulness for the running case. The algorithm used to perform this assessment is described in the next section.

Visualization methods and their selection

The process of selecting the most appropriate visualization method to display to the user needs an explicit model of the visualization methods' properties to express their specific features. This arrangement enables the process of matching the contexts against the visualization methods. A set of rules combines all these data and assigns a *total score* to each available visualization system. The property model and the selection method are described in the following paragraphs.

Visualization Method Properties

In designing a visualization system several issues are taken into account. The principal goal is to bridge the user with the information source. This is a complicated task, as many parameters have to be taken into account. Typically, the design of a visualization system has a specific focus which can be achieved using intelligent procedures. In this way, every one of the systems has its own properties, which make it unique in improving a specific aspect of the information foraging.

For example, there are visualization methods which try to display full text documents in the most effective way, using thumbnails, highlights, document size and type cues, color coding, showing relations between terms, etc. Other methods, concentrate on improving focus+context techniques, in order to give the user alternative views to the document collection, using zoom in and out functionalities, graph rotation, hyperbolic spaces etc. A very common issue in large document collections which are structured in hierarchical way is how to visualize this hierarchy in an effective and easy to explore way. Important solutions to this issue have been proposed by introducing the third dimension in the visualization design, using tree-like layouts, real world metaphors, nested items, transparency/solidity functionalities etc. One of the main concerns a Historical Archive has to deal with is the managing of temporal information, i.e. information that varies with time. To facilitate the user in retrieving such information, visualization methods employ time axes in a variety of ways: bar charts, time lines, spirals etc. Another important concern in designing a visualization method is the representation of the relation between documents. This issue is effectively addressed using links between related documents, or clustering techniques, which bring together the related documents, color coding to reveal existing relations, etc. In a similar way the problem of the representation of the relation between the query terms and the displayed results is addressed.

Finally, since a data collection is not restricted to text documents, many visualization methods focus on designing novel techniques to support users in retrieving and viewing picture, audio and/or video documents.

For the purposes of this work, visualization methods have been categorized using the classification scheme presented in (Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., & Giannopoulou, E., 2007); according to this scheme, visualization methods are primarily classified according to the visualization type they employ, which may be one of the following:

1. Indented list,
2. Node–link and tree,
3. Zoomable,
4. Space-filling,
5. Focus + context or distortion,
6. 3D Information landscapes.

Each category is further divided in two subcategories, namely 2D and 3D, taking into account the number of display dimensions it employs. While this classification scheme is introduced in the context of ontology visualization, it is generic enough to accommodate all visualization types used in this work. Using this classification facilitated the assignment of “suitability scores” for the available visualization methods, since scores were assigned at category level and were subsequently fine-tuned for each distinct visualization method within every category. The work reported in (Katifori, A., et al., 2007) includes also discussions on functional and non-functional visualization method aspects (task support, 2d vs. 3d, navigation and interaction issues and scalability issues), which have been also taken into account.

A list of basic features of visualization systems is depicted in Table 1 (the list includes only the features currently considered by the system; incorporation of additional features is being considered for future extensions). The first column lists the visualization method property, while within the second column the possible values for this property are presented. Each value is followed by an indicative list of visualization methods for which the specific property/value combination applies. Note that some visualization methods may support multiple values for a specific property [e.g. the PLAO (Lecolinet, E., Likforman-Sulem, L., Roberrt, L., Role, F., & Lebrave, J-L, 1998) visualization method may operate both in 2 and 3 dimensions], in which case the method is repeated under all pertinent list elements. Note also that in some cases, either a feature is supported or not (e.g. color coding). In these cases, no value list is provided in the second column; a dash is used instead, followed by the list of methods supporting the feature. For the compilation of the properties list appearing in table 1, a number of bibliographic sources including (Shneiderman, B., 1996), (Card, S. K., Mackinlay J. D., & Shneiderman B., 1999) and (Chi, E. H, 2000) were consulted.

Table 1. Properties of visualization systems and respective property values

Number of dimensions	• 2 (PLAO, IVEE, …) • 2 ½ (Data Mountain, LookMark, …) • 3 (IVEE, Perspective Tunnel, Task Gallery, PLAO, …)
Metaphor	• Landscape (Information City, Vineta, …) • Book and Library (WebBook, virtual library, …) • Perspective Planes & Panels (Data Mountain, Lookmark, etc) • 3D Geometric Shapes (Inform. Pyramids, VizNet, …) • Trees and Graphs (Starwalker, Visible Threads, ….)
Interactive browsing supported for documents of type:	• Article (UVA, SPIRE, Doc Cube, …) • Publication (Bead, Vineta, Cat-A-Cone, UVA, …) • Hypertext (LookMark, WebBook, …) • Photograph/Video (Viz-Net, Dynamic Timelines, …)
Supports user-defined grouping for documents of type:	• Articles (-) • Books (WebBook, Web Forager, …) • Hypertext (WebBook, Web Forager, …) • Photographs/Video (-)
Color coding Term frequency	• - (File System Navigator, Harmony Information Landscape, …) • - (Tile bars, PRISE, Themescape, …)

Visualization Method Selection

The visualization method selection procedure matches properties from the user, system and collection contexts against the visualization system properties. This matching is enabled through a rule database, containing rules of the following format:

(context-property, vis-method-property, score)

where context-property is a property from the user, system or collection context, *vis-method-property* is a visualization system property and score is a numeric metric in the range [-10, 10] expressing how appropriate visualization methods having the specific *vis-method-property* are considered for contexts where the particular context-property holds. For example, the rule

(sysctx-display-3D, vismeth-noDimensions-3, 6)

declares that visualization methods employing three dimensions are considered quite appropriate for system contexts with 3D displays, while the rule

(colctx-origin-dynamic, vismeth-itemgroup-hierarchical, -4)

expresses the belief that a visualization method employing hierarchical item grouping is inappropriate for collections that have been formulated by means of submitting queries.

For compiling the rule database, and in particular for assigning scores to (*context-property, vis-method-property*) pairs, users and visualization system experts were interviewed. In these interviews, subjects were asked to state how helpful/hindering each context property was considered in their opinion for performing each type of visualization. The interview results along with published evaluation results of visualization systems [e.g. (Robertson G., Czerwinski M., Larson K., Robbins D., Thiel D. & van Dantzich M., 1998; Shneiderman, B., Feldman, D., Rose, A., & Ferre G., X., 2000; Robertson, G. G., Van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., Thiel, D. & Gorokhovsky, V., 2000; Modjeska, D., 2000)] were used as input for the population of the rule database.

When a collection needs to be visualized, the system firstly compiles the full set of context properties, which is denoted as *CP*. Subsequently, it traverses the list of available visualization methods, extracting for each method M the set of method properties *PM*, which is used to compute a total score for method M. The total score is given by adding the score field *s* of all rules $R = (cp, vp, s)$, for which $cp \in CP$ and $vp \in PM$. Finally, the visualization method with the highest total score is selected to perform the visualization. Effectively, this step examines whether the properties of the visualization method are considered appropriate for the current context parameters, as this is expressed in the rule base. Note that under this scheme, the absence of any rule correlating a context property *cp* with a visualization method property *vp* has the effect that *vp* is considered "neutral" for contexts having the property *cp*; thus, there is no need to use rules of the form *(cp, vp, 0)* to explicitly state property orthogonalities.

An issue that has been commented on by users in the scheme above is that it is extremely prone to selecting different methods for consecutive visualization request, even though the gains (as quantified by the respective method scores) may be marginal. Since users have been found to prefer a more "stable" work environment, a provision has been added in the score calculation procedure, to increase the score for the currently used method by a value of 5. This adjustment effectively directs the algorithm to perform a visualization method switch only when considerable gains will be attained, favoring thus environment stability. The value of 5 is currently a "magic number", but in the future it is planned to incorporate it into the user context, in the sense that some users have a stronger preference towards stable environments (thus a higher "bonus" value could be used), while other users are more "adventurous", so small bonus values (or even no bonus at all) should be given to the current method, in order to pursue even marginal gains from visualization method switching.

Since for the computation of the method scores the full set of context properties *CP* is matched against the full rule set *RS*, the complexity of this operation is O(|CP| * |RS|). Note that the number of visualization methods does not appear in this formula but is indirectly considered, since the introduction of new visualization methods increases the cardinality of the rule set. Inclusion of additional visualization methods also increases the space required for storing the final result, which is O(|VM|) [where VM denotes the set of available visualization methods].

Adaptive features in method selection

The visualization method selection process described in section 4.2 does not take into account the dynamic profile of the user, as this is exhibited by the user's preferences and dislikes while working with the system. This dynamic portion of the user context is accommodated by complementing the rule list described in section 4.2 with a user-specific preferences database, which hosts information regarding:

- whether the user has considered a visualization method suitable/not suitable for a specific context.
- whether the user likes/dislikes a specific visualization method altogether.

This information is collected from the user, when the visualization task is completed (the respective window is closed) and when an alternate visualization method is requested. More specifically, the "close window" user interface widget unfolds a drop-down menu with the options "The visualization was satisfactory", "The visualization was not helpful for this data collection" and "The visualization was obscure/unusable", from which the user selects one. If the response to this drop-down is "The visualization was obscure/unusable", then the dynamic user profile is augmented with a record of the form

(dislike, viz-method)

stating that the user has a negative stance against the specific visualization method in general. Note that this does not inhibit the use of the visualization method in a future case; in presence of such rules, the visualization method selection procedure reduces the total score for the method (as described below), the method however *could* be selected if it is found to score significantly higher than other methods a specific context. If the user selects one of the two first replies, then a record of the form

(eval, system-context, collection-context, viz-method, score)

is added to the dynamic user profile, where *score* is "1" or "-1", depending on which response was selected. Note that when the user chooses one of the first two replies, the visualization method is considered helpful/not helpful *for the current context*.

The rules within the dynamic user profile are taken into account for selecting the most prominent visualization method in system context SC and collection context CC according to the following scheme:

- if a *(dislike, viz-method)* rule exists in the dynamic user profile, then the total score for the specific visualization method is decremented by 15.
- for the second form of rules, when the total score for a specific visualization method is computed the system retrieves all the rules R_{dc} = *(eval, sys-con, col-con, viz-meth, score)* pertaining to this method. Subsequently, a *similarity metric* between *(sys-con, col-con)* and *(SC, CC)* is computed, to determine which of the rules is associated with a context that best matches the current context. The value of the similarity metric falls in the range [-10, 10], with -10 meaning "totally different contexts" and 10 meaning "exactly matching" ones. The similarity metric between *(sys-con, col-con)* and *(SC, CC)* is calculated as follows:
    1. the set of all rule context facts RCF = sys-con $\cup$ col-con and the set of all current context facts CCF = SC $\cup$ CC are computed, and the similarity metric is initialized to 0.
    2. $\forall$rcf$\in$RCF, it is checked if rcf$\in$CCF. If this condition is true, the similarity score is incremented by 1, otherwise the similarity score is decremented by 1. Note that each element of RCF (and CCF) fully represents all aspects of a context element, e.g. the elements *sysctx-*

display-3D and *colctx-origin-dynamic* specify that a 3D display is used and that the collection has been formulated through a query, respectively. Therefore, if the element of RCF appears in CCF, the two contexts are identical regarding the particular context element; otherwise the contexts are different in the specific respect (and CCF would contain a different element, e.g. *sysctx-display-2D* or *colctx-origin-static*).

3. Finally, the computed similarity score ss_{val} is normalized in the range [-10, 10] by dividing by the cardinality of RCF and multiplying by 10.

Note that the context similarity computation procedure described above considers all context elements to be equally important, since any match (mismatch) contributes by 1 (-1) to the final result. Assigning different weights to context elements for the purposes of context similarity computation is an issue under investigation and will be incorporated in a future system release.

The rule with the highest positive similarity metric is finally selected, the similarity metric is multiplied by the "score" field of the rule (1 or -1, depending on whether the visualization was considered helpful or not in the specific context) and the result is added to the total score for the visualization method under consideration. If no rule has a positive similarity metric, the total score for the visualization method is not altered.

The rationale behind the computations performed using the second rule form is that if a visualization was found to be helpful/not helpful in some context, then it is "almost certain" this perception will hold for identical contexts; if, however, two contexts differ in a number of parameters, then the certainty level for this belief drops. This certainty level is reflected in the context similarity metric, while the multiplication by the "score" field simply renders the outcome positive for "helpful" visualizations and negative for "not helpful" ones.

Besides the "close window" widget, the user interface hosts the "Switch visualization" button, which provides the ability to visualize the same collection with an alternate method. In this case, the visualization methods are listed in descending order of their scores; a small sample of each visualization is presented, allowing the user to get a preview of the method before it is selected. A user may reach this decision because "An alternate view to the data is desired", "The visualization was not helpful for this data collection" and "The visualization was obscure/unusable", which are the options listed when the "Switch visualization" button is clicked. In all cases, the dynamic user profile is updated in the same way that was described for the "close window" widget.

Example

In this section we present a sample interaction of a user with the proposed system, to clarify the modeling of contexts and the score computing algorithm. In this session, the user's interaction with the system begins by submitting the free text query "Faculty of Science" to the system (the name of the faculty to which the department belongs), expecting to view tracks of the Department she is interested in, from its very beginning up to recently. The system evaluates the query against the contents of the knowledge base and determines that:

1. The query *exactly matches* a branch in the "Academic departments" taxonomy (the "Faculty of Science" branch, which is superimposed on the

Historical Archive's document collection. Though this is not a document *per se*, it is an important part of the knowledge base and is thus considered in the search.

2. The query matches metadata associated with a number of documents within the knowledge base. In particular, 136 documents have an author matching "Faculty of Science" (e.g. Dean of the Faculty of Science" and 203 documents have a recipient matching "Faculty of Science".
3. The query matches the full text of 484 documents of the knowledge base.

Subsequently, the system structures the result collection as a hierarchy, having the query as its root node, and the three subcategories identified above as its direct descendents. Each subcategory, in turn, contains the corresponding result items; the second subcategory, in particular, has an extra level of classification, separating documents whose author matched the query from documents whose recipient matched the query.

At this stage, the system has all contexts available (explicit user context, system context, implicit/dynamic user context and document collection context) and may proceed to select the most prominent visualization method. The following factors are considered (in the following, not all rules are expressly listed for brevity reasons):

1. the document collection has an hierarchical structure, so due to the rule *(colctx-structure-hierarchical, vismeth-itemgroup-hierarchical, +4)* the visualization methods that are able to intuitively present hierarchies [Cone Tree (Robertson, G., Mackinlay, J., & Card, S., 1991), the Information Pyramids (Keith A., 2000) and the Gopher VR and MoireGraphs (Jankun, T. J., & Kwan, L. M., 2003)] increment their score by 4.
2. The system has no 3D output hardware, so the rule (sysctx-display-2D, vismeth-noDimensions-3, -5) decrements the score of all 3D methods (including the Cone Tree, Information Pyramids and Gopher VR listed in the previous factor) by 5.
3. The number of direct descendents from the current root (query) is small (3), so the score of the Cone Tree is further decremented by 3 (because its the space exploitation advantage is lost in such collection contexts).
4. The score of the Cone Tree method is incremented by 4 because the user has found it useful in a situation having common elements with the current one (the recorded system and collection contexts in the respective dynamic user profile rule have some properties similar with the situation at hand) and the score of the Information Pyramids method is incremented by 2, because the contexts recorded in the respective dynamic user profile rule are less similar with the current situation.

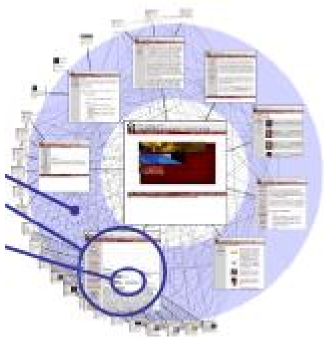

Figure 1. The Moire graph visualization

By summing up the points added/subtracted to the score of each visualization method, the system finally determines that the MoireGraphs algorithm (figure 1) achieves the highest score at this stage so it is selected for performing the visualization. In the moiré graph the query is the current focus, while the three result categories are the direct context. The MoireGraphs visualization has been tuned to display two context levels, thus the direct descendents of result categories are also shown, in smaller sizes.

Now the user focuses on the Taxonomy node, since this was the query target. The new document collection to be visualized (the taxonomy and the documents linked to each branch of it) has hierarchical structure too, but the number of direct descendents of the current root (Faculty of Science) is now considerably higher (12, which is the number of departments and administrative divisions directly subject to the faculty of science). In this respect item (3) of the factor list above does not apply, thus the Cone Tree visualization method accumulates the highest score and is selected for performing the visualization (Figure 2). In this snapshot, the Faculty of Science node is the central node in the second level, while the single node at the top level is the "University" entity. The user will locate the "Department of Informatics" node at the third level, set it as "current" and will then select "View related documents" to display all documents related with the Department of Informatics.

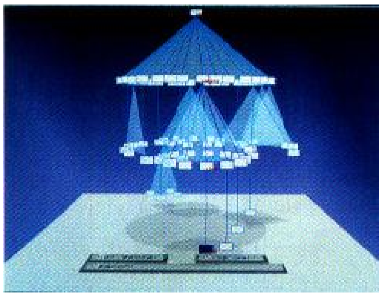

Figure 2 — Visualizing the taxonomy using cone trees

The document collection to be now presented has no hierarchical structure [property #1] and contains a considerable amount of individual documents (more than 3000) [property #2]. The first collection property adds gives an edge of 4points to the score of visualization methods not being based on hierarchies (Task Gallery, Data Mountain, Web Forager, Periscope-AVE, Virtual Library) against those who rely on hierarchical structure, due to the existence of the rule *(colctx-origin-dynamic, vismeth-itemgroup-hierarchical, -4)*. The first three of these methods are however inappropriate for large document collections (property #2), so their score is reduced; similarly, Virtual Library's score is decremented since it is more oriented to books rather than arbitrary document collections. After these computations, only the score of Periscope-AVE has not been decremented, and therefore this methods is selected for performing the visualization.

The user may finally select the desired categorization the documents (e.g. by purpose) and/or exploit the search mechanism of Periscope-AVE to locate the documents related to teaching in the Department of Informatics, which was the original goal.

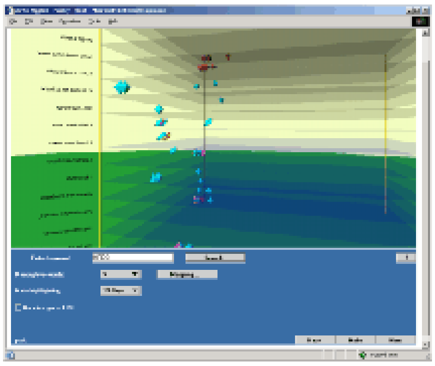

Figure 3 – Periscope-AVE visualization

Future trends

The architectural approach described in this chapter although focused on the domain of historical archives is generic enough to be used as a basis for implementing systems in other application areas, such as digital libraries, legal databases and so forth. Naturally, in any such domain, the particularities of the specific document collection will have to be studied and expressed in the format required by the score computation algorithm. Additionally, in many application areas specialized visualization algorithms have been developed; the appropriateness of any such method, in relation with any individual context parameter must be assessed and recorded in the database.

Detailed studies on if and how each context parameter affects the effectiveness of visualization methods employing certain techniques, in order to provide a more elaborate population of the rule base, are also required. A thorough system evaluation, which will provide feedback both on the overall system effectiveness and for fine-tuning the rule database, and especially the "score" field, is also needed.

Conclusions

In this paper we presented a context-based adaptive visualization environment to support information retrieval tasks in a Historical Archive. The proposed environment uses a visualization library, where a set of pre-selected visualization systems has been registered, along with their properties. Visualization method properties are matched against the task context, which includes static and dynamic user profile, system configuration and information regarding the data collection, in order to select the most prominent visualization method for the task at hand. Matching is performed through a set of rules, accommodating both generic properties (e.g. number of dimensions in the visualization, color-coding) and method-specific properties (e.g. radial graph layout).

Future Research Directions

The adaptation scheme described in this chapter relies on explicit user input, either provided during the profile population stage or given at the end of each visualization (expression of satisfaction/dissatisfaction by the user). There exist however additional information elements that can be exploited to assess the suitability and/or usefulness of a particular visualization for a specific user in a given context: these information elements can be sourced from user activity and behavior monitoring, including metrics such as idle time, use of "reset visualization" functions, erroneous activities etc. This approach requires the use of additional architectural modules, similar to the "browser/user data sensing" and "data recorder" used in (Chittaro & Ranon, 2002), while the visualization algorithms themselves need to be extended with facilities that will detect and characterize "erroneous" user activities (e.g. clicking on non-functional areas of the visualization, using "undo" operations and so forth). Generalization and abstraction mechanisms may also be introduced to allow for speedier configuration of the rule database. These mechanisms will detect common features in visualization method assessments (either explicit or derived) and formulate generic rules which will affect all methods sharing these features. For instance, if a user provides is found to dislike WebBook and Virtual library, the system may derive that the specific user does not prefer visualizations employing the "Book and Library" metaphor and thus introduce a personalization rule for decrementing the score of these algorithms.

Finally, since the number of different visualization methods, rules and user preferences within the system is expected to increase, the efficiency of the algorithm selection method will need to be addressed. Research results from the area of *efficient top-k evaluation* algorithms [e.g. (Mamoulis, N., Yiu, M., L., Cheng K. H. & Cheung D. W., 2007) and (Marian, A., Amer-Yahia, S., Koudas, N., Srivastava, D., 2005)] can be considered to this end.

References

Agrawal R. & Wimmers, E. L. (2000). A Framework for Expressing and Combining Preferences.. *Proceedings of ACM SIGMOD*, 297-306

Ahlberg, C., & Wistrand, E. (1995). IVEE: An Information Visualization & Exploration Environment. *Proceedings of IEEE Viz*

Card, S. K., Mackinlay J. D., & Shneiderman B. (1999). *Information Visualization: Using Vision to Think*. Morgan-Kaufmann, San Francisco, California.

Chi, E. H. (2000). A Taxonomy of Visualization Techniques using the Data State Reference Model. *Proceedings of the IEEE Symposium on Information Visualization 2000 (InfoVis'00)*, 69-75

Chittaro L. & Ranon R. (2002). Dynamic Generation of Dynamic VRML Content: a General Approach and its Application to 3D E-Commerce. Proceedings of Web3D Conference, pp. 145-154.

Domik G. O., & Gutkauf B (1994). User modeling for adaptive visualization systems. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceedings of the IEEE Conference on Visualization*.

Fischer G., Lemke A., Mastaglio T., & Morch A. (1991). The role of Critiquing in Coorporative Problem Solving. *ACM Transactions on Information Systems* (9), 123 – 151

Jankun, T. J., & Kwan, L. M. (2003). MoireGraphs: Radial Focus+Context Visualization and Interaction for Graphs with Visual Nodes, *IEEE Symposium on Information Visualization, Seattle, Washington*

Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C. & Giannopoulou, E. (2007). Ontology Visualization Methods-A Survey. *ACM Computing Surveys* (39) 4, 2007.

Lecolinet, E., Likforman-Sulem, L., Roberrt, L., Role, F., & Lebrave, J-L, (1998). An Integrated Reading and Editing Environment for Scholarly

Research on Literary Works and their Handwritten Sources. *Proceedings of ACM Digital Libraries Conference*.
Mamoulis, N., Yiu, M., L., Cheng K. H. & Cheung D. W. (2007). Efficient top-k aggregation of ranked inputs. *ACM Transactions on Database Systems* (32) 3.
Marian, A., Amer-Yahia, S., Koudas, N., Srivastava, D. (2005). Adaptive Processing of Top-k Queries in XML. Proceedings of the 21st International Conference on Data Engineering, 162-173.
Modjeska, D. (2000). View vs. Overview: Hierarchical Data Visualization in Desktop Virtual Reality. *Proceedings of NordiCHI 2000*.
Robertson G., Czerwinski M., Larson K., Robbins D., Thiel D. & van Dantzich M. (1998). Data Mountain: Using Spatial Memory for Document Management, *Proceedings of ACM UIST '98 Symposium on User Interface Software&Technology*
Robertson, G. G., Van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., Thiel, D. & Gorokhovsky, V. (2000). The Task Gallery: A 3D Window Manager. *CHI Letters, 2(1),* 494-501.
Robertson, G., Mackinlay, J., & Card, S. (1991). Cone Trees: Animated 3D Visualizations of Hierarchical Information. *Proceedings of CHI '91*, 189-194.
Seunghwa, L. & Eunseok, L. (2007). A Collective User Preference Management System for U-Commerce. In *Managing Next Generation Networks and Services*, Springer Berlin / Heidelberg, 21-30.
Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualization. *Proceedings of the IEEE Visual Languages*.
Shneiderman, B., Feldman, D., Rose, A., & Ferre G., X. (2000). Visualizing Digital Library Search Results with Categorical and Hierarchical Axes. *Proceedings of the 5th ACM International Conference on Digital Libraries*, 57-66.
Wiza W., K. Walczak & W. Cellary (2004). Periscope - A System for Adaptive 3D Visualization of Search Results, *Web3D 2004 Symposium - the 9th International Conference on 3D Web Technology, Monterey, California (USA), ACM SIGGRAPH*, 29-39.
Zimmermann A., Specht M. & Lorenz A. (2005). Personalization and Context Management. *User Modleing and User-Adapted Interaction*. 15, 275-302.