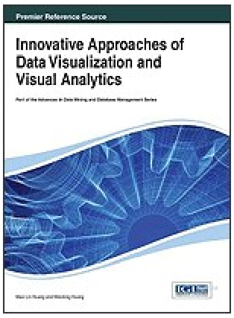

Innovative Approaches of Data Visualization and Visual Analytics

by Mao Lin Huang and Weidong Huang (eds)
IGI Global. (c) 2014. Copying Prohibited.

Chapter 14: An Information Visualization-Based Approach for Exploring Databases—A Case Study for Learning Management Systems

Celmar Guimarães da Silva,
School of Technology, University of Campinas
Brazil

ABSTRACT

Learning Management Systems (LMS) may use Information Visualization techniques and concepts for presenting their large amounts of data, in order to ease the monitoring and analysis of students learning process problems. Nonetheless, the generally adopted approaches are based on presenting data obtained by predefined database queries only, which does not consider unforeseen situations derived from final user's knowledge about e-learning domain. Therefore, the purpose of this work is to provide a resource for LMS users to define and execute queries related to these unforeseen situations. This resource is a prototype by which users may access a remote LMS database, create their own queries by selecting database attributes they want to analyze, and represent query results by means of automatically selected interactive graphical representations. User evaluations indicate that the approach is appropriate and points out possible enhancements.

1 INTRODUCTION

Providing user-level access to data stored in a database is not an easy task. It is not just a matter of providing connectivity from a system to a local or remote database. In fact, it is related also to how a user may query the database, and how he/she may interpret the returned results.

Querying a database requires that the user has gained much knowledge of database theory. This requirement includes, at least, concepts of table fields and tuples, primary and foreign keys, entities and relationships. It also requires that the user masters a database query language such as SQL. Even more, querying requires that the user knows what kind of data is stored in the database, in order to define what to query. Given these restrictions, the user's task of querying a database requires a cognitive overload which may not be dismissed, in part because typical users have not such theoretical knowledge and do not even know about the internal organization of the database to be queried. In fact, typical users are not interested in this kind of technicality, but instead they just want to obtain an answer for a question related to a system stored data.

When analyzing answers provided by a DataBase Management System (DBMS) for a query, one may think that the mostly common alphanumeric, table-based answer format is enough for answering users' questions. Even though this format is sufficient for obtaining some specific data, it may be difficult to get a data overview and to detect patterns, trends and outliers present in the data. Graphical and interactive representations of data, as proposed by the Information Visualization area (InfoVis, for short) (Card et al., 1999), are suitable for this kind of analysis. They may provide distinct levels of data overview, details on demand, and interactive capabilities of data reorganization and filtering, among other useful resources. For example, interactive filtering techniques (such as dynamic queries [Shneiderman, 1994]) afford user controls for selecting relevant data and for querying details, providing fast answers, and without the need to learn command-line querying syntax. When applied together, filtering and other techniques may enhance users' formulation of an internal model about the data under analysis.

Both difficulties–how users may query a database and how to represent query results in a useful way for them–may be analyzed from distinct scenarios, which may have distinct kinds of users with distinct computer-related and data-analysis-related skills. In this chapter, those difficulties were analyzed within a Learning Management System (LMS) scenario, in which there are students and teachers involved in learning activities. LMS are virtual environments that enable and mediate communications among participants of courses. These systems group distinct computer-based resources, like electronic mails, chat rooms, resources for publishing readings and for delivering activities' results, among others. Each resource manages and saves different kinds of data, like messages, published contents, and participant data.

In this scenario, analyzing LMS data is an important task for monitoring students' learning process, making possible to detect its potential problems. For example, participants with few accesses to the LMS, or those who do not interact with other course participants represent situations that can reveal problems related to an adopted course methodology.

Different researches try to overcome the analysis difficulty by applying InfoVis concepts and techniques, in order to graphically and interactively present LMS data. Some examples of these researches are: InterMap visual representations of course participants interaction (Romani, 2000), France et al.'s interactive activity diagram for learning scenarios (2005), and GISMO (Mazza & Milani, 2005) and CourseVis (Mazza & Dimitrova, 2005) representation of students' social, behavioral and cognitive aspects. Nonetheless, these researches show data that are obtained by predefined database queries, and so these data fit specific analysis situations. Unforeseen situations derived from final user's knowledge about e-learning domain are not considered by these researches, which do not provide ways for this user to inform the system about the data he/she wants to analyze.

Overcoming those difficulties related to querying databases and understanding query results may provide an important way for analyzing LMS data. In the LMS scenario, the presented difficulties may be summarized by two questions:

1. How to enable a user to inform the LMS what datasets he/she needs to analyze?
2. How to enable this user to get these data from the LMS, in order to analyze and understand them?

This chapter presents a proposed solution for these questions. This solution enables LMS users, each with distinct data analysis needs, to define different data combinations that they want to analyze, instead of using only system predefined queries. The solution uses database concepts of universal relation model (Ullman, 1982) for enabling users to choose database fields to be queried, without the need to know the queried LMS database internal organization. Besides, these concepts are also used for executing a query derived from this field choice in LMS database. Query results are automatically converted into interactive graphical representations by InfoVis techniques, avoiding user's concerns about developing a graphic representation that is appropriate for these data analysis. This chapter summarizes some results of Silva's doctoral thesis (Silva, 2006) and extends a previously published paper (Silva & Rocha, 2007).

The next sections are organized as follows. Section 2 presents a theoretical background on InfoVis and database universal relation model, which is necessary for understanding the proposed solution. It also summarizes how recent works apply InfoVis to LMS. The adopted solution is presented in Section 3, which describes the two systems composing the prototype–the JInfoVis and the LMS Database Explorer. This section also presents prototype's general architecture, implementation, and initial evaluation. Section 4 presents future research directions. Section 5 concludes the chapter and presents future works.

2 BACKGROUND

This section presents some theoretical background which this chapter refers to. It is organized in three parts: Section 2.1 presents InfoVis concepts used by this work; Section 2.2 presents how InfoVis is used in the context of LMS; and Section 2.3 presents the database concept of Universal Relation Model.

2.1 InfoVis

Visual mapping (Card et al., 1999) (i.e. how to transform a data table into a visual structure that users can analyze and understand) is an important process for InfoVis. Given a data table, this process represents all its data by marks into a visual structure, which can be a heatmap, a node-link diagram or a bar chart, among many others. Marks can be dots, lines, circles or other visual elements. A visual mapping associates each data table attribute (or variable) to a graphical property (color, shapes, textures, orientation etc.) or to a spatial property (e.g., its position with relation to a defined axis, or its proximity to other marks). For example, if circles are used as marks for representing users of a system, circle size may represent a variable "quantity of accesses" (how many times each user accessed this system).

Data table variables can be classified according to its semantics and its functional dependencies of other variables. Based on Card et al. (1999), Spence (2001), and Ware (2004), variables can be semantically classified into three main categories: nominal (set of elements without specific order), ordinal (set of elements with a specific order relation among them), and quantitative (set of elements with a numerical scope, and that support arithmetical operations). Considering functional dependencies aspects, each data table variable can also be defined as input or output variable, according to Spence (2001). Input variables are the variables that functionally determine the rest of data table's variables, which are called output variables.

Mackinlay (1986) points out two criteria which help defining a visual mapping for a data table: expressiveness and effectiveness. In order to achieve expressiveness, a visual structure must express exactly all data table information (i.e. all data table information, and only this information without representing additional incorrect data). Effectiveness is a criterion related to how fast a user can interpret data and easily distinguish them, with the lowest possible interpretation errors. One may compare graphical properties' effectiveness according to the data categories they represent (e.g. size is more effective than colors for showing quantitative information). Card et al. (1999, Table 1.23) present a table of relative effectiveness of retinal properties.

After representing data into a visual structure, users can interact with it in order to extract more information from it, which could not be done statically. Details-on-demand, distortions, viewpoint controls and marks rearrangement are examples of interaction techniques. Users can use radio buttons, sliders and range-sliders, among other controls, for selecting relevant data and discard irrelevant ones. In this sense, dynamic queries (Shneiderman, 1994) are an important interaction technique for users to control visual query parameters, and quickly generate a visual representation of query results. Some of their advantages are: avoiding the need of knowing command-line syntax; minimizing syntax errors when the user defines his/her query; and making it easier to construct an internal model or cognitive map about data under analysis, because query formulation and results exhibition are fast (Spence, 2001; Shneiderman, 1994).

All these and other techniques and concepts may be applied to some domain in order to represent its data and enable decision making. In this sense, next subsection presents some works that applies InfoVis to the LMS domain.

2.2 InfoVis Applied to LMS

This subsection presents a set of works that shows synergies between InfoVis and e-learning. Silva's thesis (Silva, 2006) presents additional details about some of them.

TelEduc (Rocha et al., 2002) is a LMS that uses some InfoVis related techniques in order to enable a distinction among "silent and present" students and really "absent" ones, a needed differentiation for tracking a full-distance course. In this sense, the InterMap (Interaction Map) tool (Romani, 2000) uses node-link diagrams, heatmaps and bar charts, among other visual structures, for graphically representing course participants interaction into TelEduc communication tools (Mail, Chat, and Discussion Forums). A similar objective lead TelEduc development team to create a tool called Access (Silva & Rocha, 2004), which stores and presents course participants accesses to the course starting page and to the TelEduc tools available for that course. TelEduc is an open source LMS that has been developed since 1997 by the Nucleus of Informatics Applied to Education and the Institute of Computing, both at University of Campinas.

Access and InterMap tools highlight questions that are also addressed by Hardless and Nulden (1999). According to them, e-learning-based course instructors report difficulties in perceiving what is happening in their courses. Hardless and Nulden propose a software called Activity Visualization, which aims to reduce this kind of "blindness". It uses heatmaps for showing course participants' access and messages along the time. However, it is not possible to conclude from their paper whether their software provides an interactive heatmap, neither what LMS it was connected to.

GISMO (Mazza & Milani, 2005) and CourseVis (Mazza & Dimitrova, 2005) represent Moodle and WebCT course data, respectively. Both use heatmaps and histograms for representing students' social, behavioral and cognitive aspects, such as: data about participants' accesses to LMS courses or resources (contents), data about tasks and quizzes, and data about participants' discussions. GISMO also provides interactive controls for manipulating the visual structure (for time interval selection and for selecting students, groups, resources, etc.).

Mazza (2006) presents an interactive system that enable Moodle administrators to compare the intensity of access to Moodle tools in many offered courses. It provides a representation based on pixel-oriented techniques, and enables administrators to select courses, time intervals and aggregation unit (daily, monthly, etc.) for access count.

Tzoumakas and Theodoulidis (2005) focus not just on visual representations of course participants' interaction, but also on representations of interactions between students and resources, and between students and software. Force-directed and radial graphs represent these interaction types, whose data is provided by WebCT e-learning environment logs.

Saltz et al. (2004) propose visual structures for helping instructors to perceive students participation in online discussions and to detect students with difficulties for creating an active dialog. The proposed solution, called "social student graph", is a node-link diagram whose nodes are students and whose directed edges represent messages sent to a specific student under analysis. Node size and form represents, respectively, the amount of sent messages and the participant role in the course (instructor or student). The proposed visual structure, however, does not have interactive features neither is connected to a LMS.

Otsuka and Rocha (Otsuka, 2006; Otsuka & Rocha, 2007) use non-interactive graphics for helping instructors to assess LMS students participation in a formative assessment process. Their multi-agent system represents different LMS data, such as: relevance of messages sent by course participants; students participation regularity in an activity; amount of messages and comments sent by participants; time of activity delivery by students (in relation to activity deadline); and students' performance. All these data is represented visually as bar charts, line charts and pie charts.

Foroughi and Taponecco (2005) propose bar charts and spiral-based visual structures for representing and contrasting attributes such as time spent by students to resolving problems and students' performance. These graphics present interactive features such as zoom, rotation, brushing and navigation through the available visual structures. Gómez-Aguilar et al. (2009) also use an interactive spiral timeline; their spiral present a histogram of activities (among other elements) along the time. It is improved by background bars that present the intensity of activities in higher units of time than the used by the histogram. Filtering and alternative views of data are also available.

France et al. (2005) use simultaneous representation of an activity diagram for a learning scenario and a set of LMS-related logs, in order to enable student behavior analysis. Their system employs focus+context techniques for representing these logs.

The aim of Nguyen, Huang and Hintz works (Nguyen & Huang, 2005; Huang et al., 2005) is helping to understand data stored in LiveNet collaborative learning environment. They focus on representing relationships among different collaborative objects, such as activities, contents, and participant groups. These relationships are presented by means of hierarchical representations based on encapsulation and connection techniques used simultaneously.

Klerkx et al. (2004) study how to use InfoVis techniques for easing object search into a learning content management system. Different search strategies are enabled by InfoVis techniques for representing hierarchical data, such as the above mentioned encapsulation and connection ones.

Hijón-Neira & Velázquez-Iturbide (2008) use a subset of Prefuse visualizations for representing diverse LMS data, including: a scatterplot showing the evolution of the amount of work done by students; a fisheye-based textual description of student accesses; a node-link diagram that groups students according to their grades; a Prefuse's "data mountain" representation with the temporal evolution of accesses; and a "data mountain" with characteristics of students' computing equipment and geographical location.

Hijón-Neira et al. (2008) use Spotfire for analyzing students interactions with a LMS (Merlin), trying to correlate access patterns, course period (morning or evening) and exam performance. They neither present how data was transferred from LMS to Spotfire, nor the characteristics of the users who used Spotfire for analysis.

Martín et al. (2011) presented a module of a LMS whose pie charts present percentages of time spent by users for concluding exercises and their correctness. These exercises may be selected by an outline. They point out that other visual representations are available but their paper does not present them.

Teutsch and Bourdet (2010) state the importance of visualizing three dimensions of LMS data: participants, calendar (time) and scenario (the course structure itself, composed by tasks, content and methods of participation [activities]). They propose to analyze pairs of these dimensions, represented as axes of a kind of scatterplot. Another dimension may be represented by mark colors.

France et al. (2006) propose a visualization system which represents three views: classroom, student and activity view. They use Chernoff faces for representing students: face characteristics are mapped to the number of student logs and the time spent by the students doing some exercise. In "classroom view", Venn diagrams group students that are doing the same activity, and colors indicate group delay for accomplishing the activity. The "student view" present a scatterplot of number of logs accumulated per minute by a student in order to complete course activities; this view also compares this number with an histogram of average number of logs from other students in the same activities. The "activity view" presents a scatterplot correlating time spent per each course participant for accomplishing an activity, and average time of a student relative to the classroom average time of doing exercises. It aims to help teachers to analyze if students are spending more time than usual to conclude an activity.

Gómez-Aguilar et al. (2008) uses Prefuse for presenting some visual representations for LMS elements (participants, logs and data stored by Moodle tools), such as a node-link diagram connecting these elements and a word cloud about forum messages. A more recent work of Gómez-Aguilar et al. (2011) presents a tag cloud integrated to a wave-graph and a bar-graph for representing the frequency of words in LMS tools along the time.

Gómez-Aguilar et al. (2010) surveyed some InfoVis works related to the use of visualization for representing the learning content and learning objects, for organizing and managing learning objects, and for representing the learning process in order to better understand it.

A comparison of the characteristics of these works reveals a common aspect: each one deals with visualization solutions implemented for helping users with specific analysis needs. This way, users whose needs were not contemplated by the previous works still have arduous effort for collecting LMS data from the available user interfaces and for organizing them into a chart, so as to better understand the data. Section 3 presents a possible solution for this problem. Before it, the next subsection introduces the Universal Relation Model for a better understanding of the remainder of the chapter.

2.3 Universal Relation Model

As the introduction section states, this work aims to enable users to query a LMS database without the need to know its internal organization. This objective is shared with some Database researches, which aim to isolate users from the need to know the data's logical structure (such as tables, registers, entities and relationships). These researches use the concept of universal relation model for doing this kind of logic isolation. A database universal relation is a hypothetical relation whose schema is composed by all attributes from all relation schemas of a database (Ullman, 1982). For the user who defines the query, it seems that the database has just one big relation (the universal relation) to which he/she may send queries. For the DBMS that supports this model, it is necessary to convert each universal relation-based query into a query that can be executed in the relational model. This conversion mechanism, called query interpretation, frequently needs to define lossless joins that were not defined in the original query. This join definition is the heart of query interpretation algorithms, and defines the sense of the query and, consequently, the sense of its result.

3 PROPOSED SOLUTION

The presented background on InfoVis and Database enables constructing a solution for the LMS querying problem, stated at the beginning of this chapter. It is possible to help LMS users to obtain answers to questions related to its analysis, in two ways:

1. Enabling users to define their queries as a set of LMS database attributes, selected by the user among all table attributes, and without need to know the database internal organization. Query interpretation algorithms may transform the selected attribute set (considered as a universal relation query) into a relational query, which enables it to be executed in LMS database.
2. Enabling them to visualize graphically and interactively the query results, helping them to analyze and understand these results, and to get the desired answer. This solution also determines that the system itself, instead of the user, can define suitable graphical representations for the results.

This solution was divided into the following six steps:

1. The system shows LMS database attributes, so users may select some of them. Database tables are not shown for users, because they must see the database as a single universal relation. Attributes derived by applying aggregate functions (quantity, maximum, minimum, sum and average) to other database attributes may also be selected. All attributes are shown with descriptive names instead of their original names.
2. After the user selects a non-empty set of attributes, the system analyzes the selected attributes and verifies how functional dependencies relate them in the database. The observation of primary, alternative and foreign keys in the database schema infers these functional dependencies.
3. Based on the selected attributes, on their categories, and on their functional dependence-based relationships, the system chooses a visual structure (among the available ones) that it understands as a best-fit choice for showing query results, taking into consideration expressiveness and effectiveness concepts. In this step, user can go back to step 1 and modify the selected attribute set, or else he/she can continue to the next step.
4. Based on the selected attributes, the system prepares and executes an SQL query for getting the asked data. This step uses universal relation theories for join inference if necessary.
5. After query execution, the system uses the system-selected visual structure for showing the obtained data.
6. The system enables the user to explore the visual structure by means of dynamic query-based controls and functionalities. The user may reorganize data, filter them and ask for details about them.

In order to provide this solution, a prototype was planned and implemented. This prototype has two main parts: LMS Database Explorer, which provides attribute selection and query interpretation capabilities; and JInfoVis, which provides graphical and interactive representation of results, and which is used by LMS Database Explorer. Both prototype modules are described in the following subsections.

3.1 LMS Database Explorer

The Learning Management System Database Explorer (or LMS Database Explorer, for short) is a system which aims at giving users the capability of querying a LMS database even if they know neither database concepts nor the database itself. It is related to all six steps of the presented solution, but specially to the former three, which will be detailed in this section. Section 3.2 will detail the latter three, which are more related to JInfoVis.

3.1.1 Showing Database Attributes

In order to enable users to choose a subset of the database attributes – the first step of the solution – the system must show all those attributes. However, the attributes names may have often no meaning for user, even if concatenated with their table names (e.g., Mail.message_id). Attribute names must then be converted into meaningful and unambiguous descriptions for enabling users to select attributes which are appropriated for their queries.

Grouping attributes by some subject is also necessary for helping users find what attributes they may select. This work considers that users have experience in using a LMS, and then the LMS concept of *tools* (Mail, Discussion Forums, Whiteboard, Support Material etc.) should be well-known. Based on this fact, it is proposed that the system groups LMS attributes by tools.

The attribute selection step also must take care of ethical and security-related questions. If all database attributes are available for users' queries, they would be able to acquire login and password data from other users. Another equally undesired situation is that users would find it possible to obtain other users' private data, such as personal messages and contents stored in the LMS. Blocking users to select specific attributes such as login, password and mail messages helps to solve this problem.

Sometimes it is also necessary to summarize the values of some attributes. For example, the analyst may not be interested in each forum message, but in the quantity of forum messages. Another example is the observation of when the users' most recent accesses to the LMS happened, and not each one of their accesses. Taking this into consideration, it is necessary to provide users the possibility of selecting values calculated by aggregation functions, such as sum, average, maximum value and minimum value. This chapter calls this kind of pseudo-attributes *derived attributes*, which are actually result columns of aggregation-function-related queries. For differentiation purposes, attributes that are originally available in the database will be called non-derived attributes.

3.1.2 Analyzing Selected Attributes

Given that a user has selected a set of attributes, the system must understand some characteristics of these attributes in order to propose a graphical and interactive representation, which will represent the dataset related to them.

The first characteristic to consider is the semantic category of the attributes, a concept already presented in Section 2. Database schemas define attributes in terms of data types (integer, float, string etc.), which are not sufficient for defining the data semantic category. Hence, it is necessary to provide this kind of metadata for each database attribute.

Beyond classifying database attributes, it is also necessary to categorize derived attributes, given that users may select them. In order to address this problem, one should first consider which aggregation functions may be applied to which attributes, given that their semantic categories may block the use of some functions. As an example, it does not make sense to apply an average function on nominal values, but one may apply it on quantitative values without problem. The latter situation will produce a derived attribute which is also quantitative. Therefore, given the categories of each aggregate function input attributes, it is possible to state what the output attribute category is, as presented in Table 1.

Table 1: Categories of aggregation functions' output attributes, according to input attribute categories. Adapted from Spence (2001, p. 68, Tables 5.1 and 5.2)

Aggregation Functions	Input Attribute's Categories		
	Nominal	**Ordinal**	**Quantitative**
count	quantitative	quantitative	quantitative
min, max	(unavailable)	ordinal	quantitative
sum, avg	(unavailable)	(unavailable)	quantitative

Another analysis one must do before choosing a visual structure is how the selected attributes are interrelated. This relationship may be defined in terms of *functional dependencies*, a well known concept of database area. Consider a database relation R and two attribute sets of R, called X and Y. If one verifies that $t_1(Y)=t_2(Y)$ for any couple of tuples t_1 and t_2 for which $t_1(X)=t_2(X)$, then this situation characterizes that X functionally determines Y (which is denoted by X→Y). This concept of functional dependency may be transitive: if A→B and B→C, then A→C, according to Armstrong's transitivity rule (Elmasri & Navathe, 2000). Therefore, a well defined database may be understood as a directed graph, whose nodes are non-derived attributes and whose edges are functional dependencies.

In order to consider derived attributes and universal relation model, one may apply also the following functional dependency rules:

1. An attribute set functionally determines each of its attributes (which is derived from Armstrong's reflexivity rule [Elmasri & Navathe, 2000]).
2. A derived attribute functionally determines only itself.
3. A primary or alternative key from a table functionally determines other table attributes (which is derived from the concept of key).
4. Consider a foreign key F which points to a primary or alternative key K. Following a universal relation model, each attribute of F merges with an equivalent attribute of K through a natural join. Therefore, F→K and K→F.
5. Consider a derived attribute D, which results from an aggregation function applied to an attribute set A. Therefore, A→D. Observe that D does not exist in the real database, and consequently this functional dependency only may be defined after query definition.
6. Let A and B be two distinct attributes, such that A→B and ¬(B→A), either transitively or not. Therefore, there is a value of B for each

value of A, and many values of A may exist for each value of B. Let D be a derived attribute, such that A→D (as presented by the rule #5). Therefore, there is a value of D for each value of A. From both situations, many values of D may be indirectly related to each value of B through A. Consequently, grouping values of B by values of D make sense, and then B→D.

Rule #6 deserves an example. Suppose that class, teacher and student are some universal relation attributes, such that *class→teacher* and ¬ *(teacher→class)*. Suppose that *class→number_of_students* (a derived attribute generated by a count function applied to student). Given that each teacher possibly teaches more than a class and each class has a number of students, then each teacher teaches a number of students. Therefore, there is a functional dependency *teacher→number_of_students*.

The presented rules, together with functional dependency original ones, enable the system to calculate functional dependencies from non-derived and derived attributes. As the following subsection presents, system uses these dependencies for defining which visual structures to use.

3.1.3 Choosing Visual Structures

The previous step analyzed some important characteristics of selected attributes, nominally functional dependencies and semantic categories. Based on these characteristics, the system must propose a visual structure for representing data. Among all available visual structures, the chosen structure must be one that better fits data, that is, one with the highest effectiveness and expressiveness related to attributes semantic categories and functional dependencies.

The available visual structures considered in this work are bar charts and heatmaps (which may be considered as starfield displays or even double-entry tables), and they are available for use in JInfoVis. Given these visual structures, it is necessary to define which structure to use, and how to do the visual mapping among its graphical properties and selected attributes.

First, it is necessary to understand that *each available visual structure has its own expressiveness*, that is, a capability of expressing exactly a specific type of input data. Consequently, expressiveness criteria restrict which datasets may be represented by which visual structure. For example, a bar chart, as implemented by JInfoVis, has a horizontal axis H which express nominal or ordinal values, and a vertical axis V which express only quantitative values. Besides, for each value of V there is only one corresponding value of H. It is equivalent to affirm that H→V.

An equivalent analysis may be applied to heatmaps. The heatmap implemented by JInfoVis has a horizontal axis H and a vertical axis V. Both may express nominal, ordinal or quantitative values, and a functional dependency between H and V is not necessary. Optionally, a legend L may exist, and it may represent quantitative or ordinal values (other heatmap implementations may also support nominal values). JInfoVis heatmap represents a single output value for a pair of input values, one from H and other from V. Therefore, it has a functional dependency HV→S. Figure 1 represents thumbnails of the three presented possibilities of visual structures.

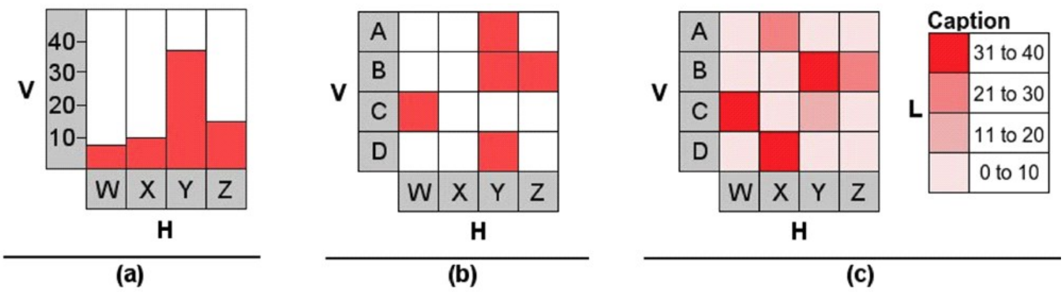

Figure 1: Visual structures: (a) bar chart, (b) simple heatmap, (c) heatmap with legend

The presented analysis is related with visual mapping, which maps data table variables into graphical properties. However, one must think about *how database selected attributes are related to these variables*. A first and intuitive approach is to associate directly each attribute to a variable. Consequently, this solution works as if each attribute was mapped to an axis or to a legend.

A second and a little bit less intuitive approach is to map more than an attribute to a single variable. Suppose that attributes $A_1,\dots, A_n$ are mapped to data table variables $V_1,\dots,V_n$, respectively. Besides, suppose that a data transformation originates a new data table, in which V_1, …, V_n are demoted to values of a new variable W. All cases of the previous data table are then redefined, in such a way that values of each variable V_i are merged into a single value. W is represented as an axis in the visual structure, which consequently shows all attributes $A_1,\dots,$ A_n. Figure 2 exemplifies this second approach. It is important to highlight that all attributes $A_1,\dots, A_n$ must be similar enough to enable demotion; for example, all attributes may be quantitative ones.

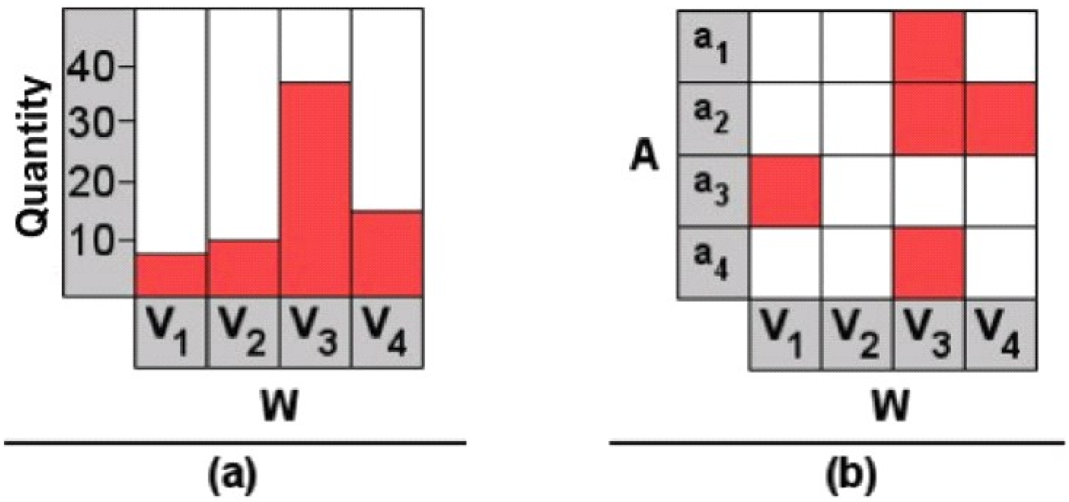

Figure 2: Multiple variables (V1,…,Vn) are mapped to a new variable W. In (a), each variable Vi determines a quantitative value (the variable "Quantity") In (b), {Vi,A} determines a variable represented by color

Given the functional dependencies among user-selected attributes, semantic categories of attributes, expressiveness of available visual structures and relationship between attributes and data table variables, it is possible to verify if there are available visual structures able to represent the results from the user-defined query. If there is more than one structure available, it is also possible to choose the one that better fits the data.

When considering expressiveness criteria, a visual structure is able to represent the results of a query if two conditions are obeyed. The first one is that a subset (S) of the attribute set functional dependencies (A) must match the set of visual structure functional dependencies (F), in order to provide a minimally necessary expressiveness. The more S and F matching functional dependencies, the higher the expressiveness. For example, suppose that {X→Y} is the set of functional dependencies of the attribute set {X,Y}. Bar charts' functional dependencies set is {H→V}, which matches exactly {X→Y} if X is mapped to H and Y is mapped to V. The heatmap functional dependencies set for a two-variable case is an empty set, but this does not block a heatmap from representing X and Y relationship. In this case, both bar chart and heatmap may represent the attribute set {X,Y}. However, the bar chart is more expressive than the heatmap in this case, because the bar chart expresses the X→Y dependency, but the heatmap does not.

The second expressiveness condition is that a data table variable (which may represent a single database attribute or a set of attributes) only may be mapped to a graphical attribute (axis or legend) that matches its semantic category. In the previous example related to bar charts, X must be nominal or ordinal (because H has this restriction), and Y must be quantitative (because V has this restriction). If these characteristics do not happen, it is necessary to use the next available visual structure, even if it is less expressive than the former.

Effectiveness must also be considered for defining a visual mapping. Concerning this, this chapter presents an approach that prioritizes the use of spatial position of marks in the visual structure. This approach defines that input variables have priority to be mapped to spatial axes. This characteristic privileges the visual comparison of data related to distinct values of a same input variable.

Another feature of the proposed effectiveness approach is related to heatmap legend colors. Legend represents an output ordinal variable. Consequently, values of a quantitative attribute must be classified in order to be represented by legends, and therefore this classified attribute becomes ordinal. Distinct color sets are used for distinct variable categories. Quantitative or ordinal variables are mapped to color brightness, and hue and saturation are kept constant. Ordinal temporal variables, however, uses new-leaves color for representing recent values, and old-leaves color for old ones. The former decision is based on the effectiveness of bright for representing quantitative and ordinal values (Card et al., 1999). The latter is based on Tufte's work (Tufte, 1990), which suggests using colors found in nature, given our familiarity with them, their coherence and harmony.

Taking in consideration these expressiveness and effectiveness criteria and all the definitions presented in this chapter, the following algorithm chooses a visual structure and a related visual mapping for a given set of user-selected attributes.

1. Define X as the user-selected attributes.
2. Identify all sink attributes. A sink attribute is an attribute functionally determined by all other user-selected attributes, transitively or not.
3. If X has just a single sink attribute, this attribute will be mapped to an output variable, because it is defined by all other selected attributes. In this case:
    a. Lasting attributes will be mapped as input variables.
    b. If there is only one input variable (and therefore only two attributes were selected), two situations may happen:
        a. If the output variable is quantitative and the input variable is ordinal or nominal, create a bar chart. Its horizontal and vertical axes must represent the input and output variables, respectively, which respects the bar chart restriction H→V. (End.)
        b. Otherwise, create a heatmap without legends. Its vertical and horizontal axes must represent the input and output variables, respectively. (End.)

c. If there are 2 input variables, and the output variable is not nominal:

   a. Create a heatmap with legend. Its horizontal and vertical axes must represent both input variables, and the legend must represent the output variable. This configuration obeys the heatmap restriction HV→L. (End.)

d. Otherwise, there are not visual structures available, suitable for representing query results. (End.)

4. If X does not have a sink attribute, only one of the following situations must happen:

   a. If all X attributes are quantitative, create a bar chart. Merge all X attributes into a single input variable. Map this variable to the bar chart horizontal axis. The vertical axis represents quantities related to each X attribute, which constitutes an output variable generated by merging. (End).

   b. If all X attributes are ordinal temporal, create a heatmap without legend. Similarly to the previous situation, merge all X attributes into a single variable. However, map this variable to the vertical axis instead of the horizontal one. The horizontal axis must be mapped to time, which constitutes an output variable generated by merging. (End.)

   c. Consider an attribute A such that A ∉ X. If all X attributes determine A, create a heatmap without legend. Map A to an output variable, which must be mapped to the horizontal axis. As in previous steps, merge X attributes into a single variable, and map this variable to the vertical axis. (End.)

   d. Consider that X has only 2 selected attributes, A and B, and that there is an attribute C, which functionally determines both. In this case, take A and B as input variables. A new binary output variable O is defined by a function f, defined as follows: given a ∈ A and b ∈ B, define f(C,a,b)=true if there exists c ∈ C such that a, b and c share at least one of the universal relation tuples; otherwise, f(C,a,b)= false. Create a heatmap without legend, whose axes map A and B. The output variable O determines the existence or absence of marks in each possible coordinate of the heatmap.

   e. Otherwise, there are no available visual structures suitable for representing query results. (End.)

5. Otherwise, if there are more than one sink attribute, there are no available visual structures suitable for representing query results. (End.)

Figure 3 summarizes this algorithm. It is important to highlight that the presented algorithm considers only heatmaps and bar charts as available visual structures. Future extensions of this algorithm must consider other possible visual structures.

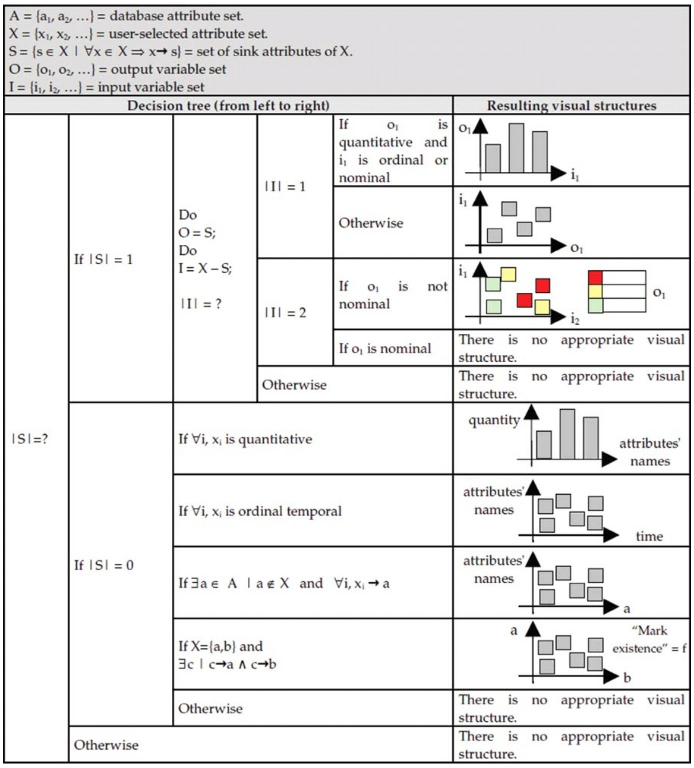

Figure 3: Summarized decision tree for choosing a visual structure

3.1.4 Alternative Visual Structures for Selected Data

Shneiderman (1996) states in his Visual Information Seeking Mantra–"overview first, zoom and filter, then details on demand"-a data navigational approach by which users should first see a overview of the data and then navigate through the details. JInfoVis approach agrees with this mantra, but with a slight different way of providing details.

Among the visual structures provided by JInfoVis, heatmap is the one that represents more variables at the same time (three, when using legends), and then it is the most general one. If a user asks for details of a heatmap line or column, the result for this kind of query also must be effectively represented. Suppose that A, B and C are three variables visually mapped to the vertical axis, horizontal axis and legends of a heatmap, respectively. If the user asks details about a value $a_1 \in A$, he/she is not interested in other values of A, and therefore A does not need to be represented at the heatmap. Removing A from it, there is a free vertical axis at the heatmap. In order to provide a more effective visual mapping of data, C should move from legends to this vertical axis. Variable A may be mapped to a single-value selection filter, closer to the heatmap, and with a_1 as a preselected value. If C is an ordinal or nominal variable, the previous solution fits well. However, if C is a quantitative variable, it is more expressive to represent B and C through a bar chart instead of a heatmap. Figure 4 shows this situation.

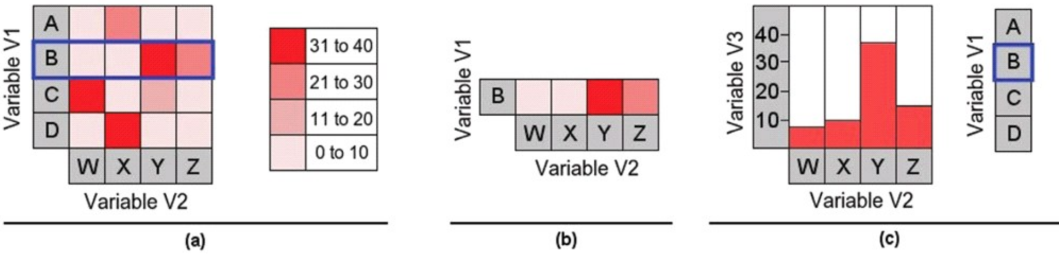

Figure 4: Example of accessing more detailed views of data (a) A heatmap representing two variables by axis (V1,V2) and a third one by legend (V3) A user asks details about the selected line (value B of variable V1) (b) After filtering, only value B is present at the Y-axis (c) V1 is removed from the graphic itself and is placed besides it, at a selection control This control shows the selected value B V3 may use the dismissed Y-axis for representing its quantitative values Therefore, the visual structure becomes a bar chart with a coupled selection control

Indeed, this kind of operation does not reveal more data (and, therefore, should not be called a "detail-on-demand" operation), but it aims to enable a correct visual comparison of the selected data.

As presented, LMS Database Explorer defines a visual structure (and its related visual mapping) for representing user's query results. JInfoVis, presented in the following section, is responsible for executing queries, drawing visual structures and enabling user interactions.

3.2 JInfoVis

JInfoVis is an Information Visualization toolkit prototype. It provides infrastructure for querying a database and for showing visual structures and associated widgets for users. JInfoVis was created in the context of LMS data analysis, aiming to ease observation of data relevant for analyzing user behaviors and for course-related decision making. Despite this, JInfoVis may be used in other data analysis contexts which are not related to LMS.

JInfoVis was conceived according to a set of principles defined by the authors' previous experience on using InfoVis concepts at TelEduc, and by analysis about more intense use of InfoVis techniques for visually representing LMS data. These principles, considered as requirements for defining JInfoVis, are presented in the following subsections. Some considerations about how to show attributes for users are also presented.

3.2.1 InfoVis-Related Characteristics

The following InfoVis-related requirements determined JInfoVis development:

- Avoid excessive use of numeric representation of data, in order to reduce users' cognitive effort for data comprehension.
- Enable users to access different levels of data generalization, as presented in Section 3.1.4. JInfoVis does not define what visual structure may represent details about data selected in another visual structure. However, it enables programmers to define what happens when the user asks details about a subset of the presented data, and here programmers may invoke a second visual structure or any kind of detail information.
- Also according to Shneiderman's mantra, provide for users filtering capabilities, such as dynamic query filters (Ahlberg & Shneiderman, 1994) and direct selection mechanisms, in order to hide irrelevant data.
- Provide data reordering capabilities for users, obeying visual structure organization. Spence (2001) points out that this kind of rearrangement may provide new insights about data.
- Use simple visual structures, with few variables, in order to ease data understanding. Given that heatmaps may represent two or three variables, and bar charts two variables, these are the visual structures implemented by JInfoVis, as already mentioned in this chapter. Node-link diagrams are also implemented by JInfoVis, but they were not used in the context of this work.
- Ease dataset comparison by users, avoiding motor and cognitive overhead. In order to accomplish this, the following sub-requirements (related to direct manipulation concept) must be attended:
    - Enable users to define query parameters about data to be analyzed, and provide in the same query window the data resulted from query execution.
    - Provide responsive interaction (Spence, 2001), reflecting instantaneously into the presented dataset each query parameter change.

  - Provide query parameter change by user interaction through keyboard (e. g. entering a data interval) or by pointing devices (e. g. dragging a range slider).
  - Provide undo and redo capabilities.

3.2.2 Query Execution Requirements

One of the JInfoVis responsibilities is executing queries in the LMS database. The main requirements related to this task are:

- Allow for querying LMS database every time it is necessary–and not just at application startup – because JInfoVis interactions may demand more data after presenting a visual structure.
- Avoid querying LMS database frequently.
- Provide secure and remote querying capabilities, because LMS database will not be stored in the users' computers, but in a remote server.
- Allow for the execution of interpretation algorithms for querying LMS databases according to the universal relation model. An interpretation algorithm related to an attribute set X must determine a set of lossless joins among the database relations which contain X. This requirement is an LMS Database Explorer demand.

3.2.3 Preparing Data for User Comprehension

In order to enable users to understand a dataset, it is relevant to prepare data before presenting them. This is necessary because a database may have attributes and values that make sense only for database and system designers, and not for system users. Concerning this, this section presents situations which must be taken into consideration for enhancing user understanding about data.

The already presented concerning about showing attribute names (Section 3.2) also applies to the time when attributes will be presented by JInfoVis. This may happen when many attributes are mapped to a single variable, and hence attribute names are presented at the visual structure axes.

This only-understandable-by-programmers problem related to attributes names also applies to attribute values. These values may be system-defined codes related to system states (e.g., single-character attributes such as "N," "R," and "A" are not directly meaningful for users but, for a mail system, they may represent new, read and answered messages, respectively). Meaningful descriptions of these values must be presented for users instead of the original values, in order to enhance their understanding about data. Similarly, the attributes null values also may have an implicit meaning that must be clarified for the user.

Substitute keys require also some attention. A substitute key (defined by Date (1986) as "substitute") is a system-defined immutable single-attribute primary key, e.g. an auto-increment attribute. Once its values only matters for the system itself and not for users, a simple approach (A) is avoiding users to select this kind of attribute. A second approach (B) is presenting each of these attributes for users (because they represent a single database table tuple), but showing values of another attribute which is more relevant for users than the primary key artificial codes. For example, suppose a database table *User(user-id, name, address, phone-number)*; if a user asks for user-id values, the system may show name values instead, which are more descriptive than user-id codes.

Foreign key attributes that point to substitute keys also have system-defined values, and hence they may be transformed into descriptive values by the same A and B approaches, previously presented. (E.g. if a user asks for user-id values of the table *Participation(user-id, class-id)*, and Participation.user-id is a foreign key that points to User.user-id, the system may present User.name values instead of Participation.user-id ones, according to B approach). Indeed, B was the approach implemented by JInfoVis.

In order to meet all of the stated problems, JInfoVis must have access to LMS database metadata which must hold descriptions of attributes values and names. JInfoVis must also access LMS database schema in order to analyze tables' foreign and substitute keys.

3.3 Prototype architecture

Both JInfoVis and LMS Database Explorer were implemented according to the architecture presented in Figure 5. In this architecture, the LMS database and its metadata are stored in the LMS server. A JInfoVis module called Request, which mediates the acquisition of data from the database, is a servlet that also resides in the LMS server. The remaining system runs in the users' web browser as a Java applet.

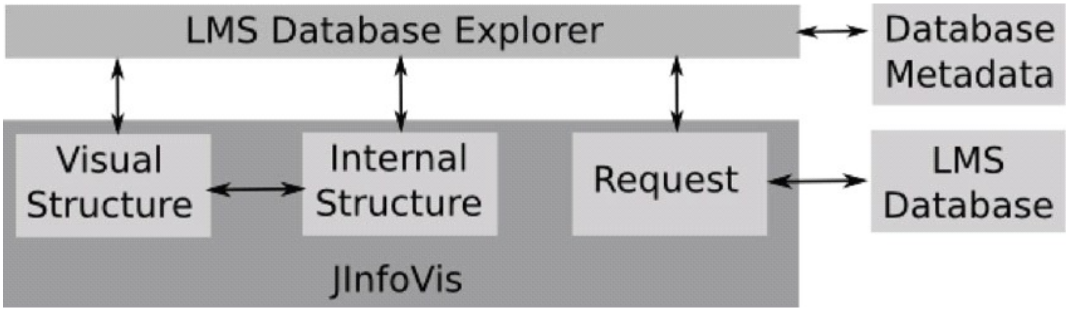

Figure 5: System architecture

According to the presented architecture, LMS Database Explorer initially gets LMS database metadata. After a user asks a query, LMS Database Explorer interacts with JInfoVis (through Request module); the former informs the latter about the query to be executed in the LMS database. The request module executes the query, obtains its results and feed them back with the information. When LMS Database Explorer receives these results, it stores these data in the Internal Structure module, which stores the data into a specific data structure. Finally, LMS Database Explorer uses Visual Structure module for defining what visual structure will be connected to the stored data, that is, which data must be shown by the defined visual structure. User interactions may ask more details about a data subset. LMS Database Explorer answers this event with another visual structure, which demands a new dataset. Hence, the request-and-show cycle restarts.

The system uses TelEduc (Rocha et al., 2002) as its subjacent LMS, and accesses its database and its related metadata. Given that neither JInfoVis nor LMS Database Explorer depends on TelEduc implementation, the overall system works also with another LMS whose data persistence layer is managed by a relational DBMS. Future developments may connect the implemented system to another LMS, such as Moodle and TIDIA-Ae.

3.4 Implemented Prototype

The prototype system was implemented according to the previously presented characteristics, and the following figures depict it. In Figure 6, a user executes LMS Database Explorer. This figure shows a list of LMS database attributes related to the subject Discussion Forums ("Fóruns de Discussão", in Portuguese). He chooses the following attributes: "discussion forums messages date", "discussion forums", and "discussion forums messages – quantity". At the bottom of the figure, the system informs that a heatmap is available for interconnecting the selected attributes values.

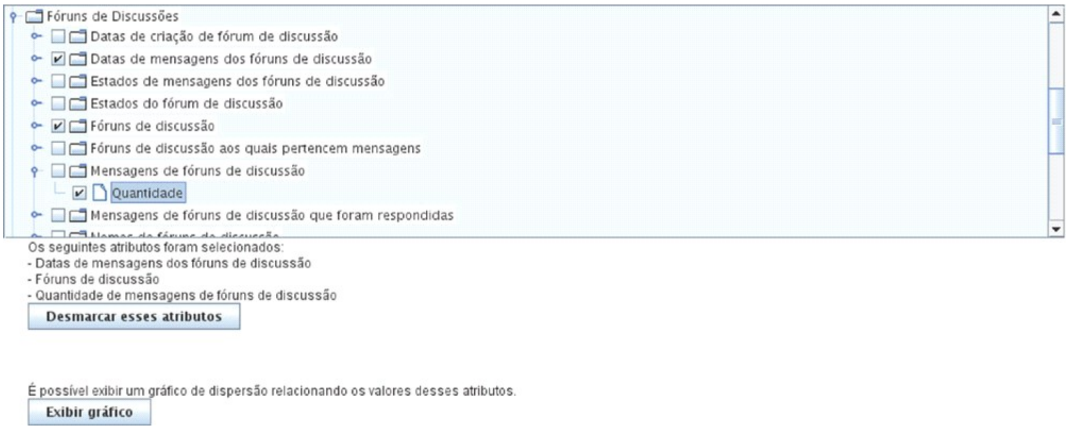

Figure 6: User selecting LMS database attributes

When user press the "Show graph" ("Exibir gráfico", in Portuguese) button, the system runs a query on the LMS database and then shows Figure 7, which illustrates JInfoVis in action. It shows the selected database attributes as elements of a heatmap, which presents the quantity of forum messages (represented as color) by forums (at y-axis) and by message date (at x-axis).

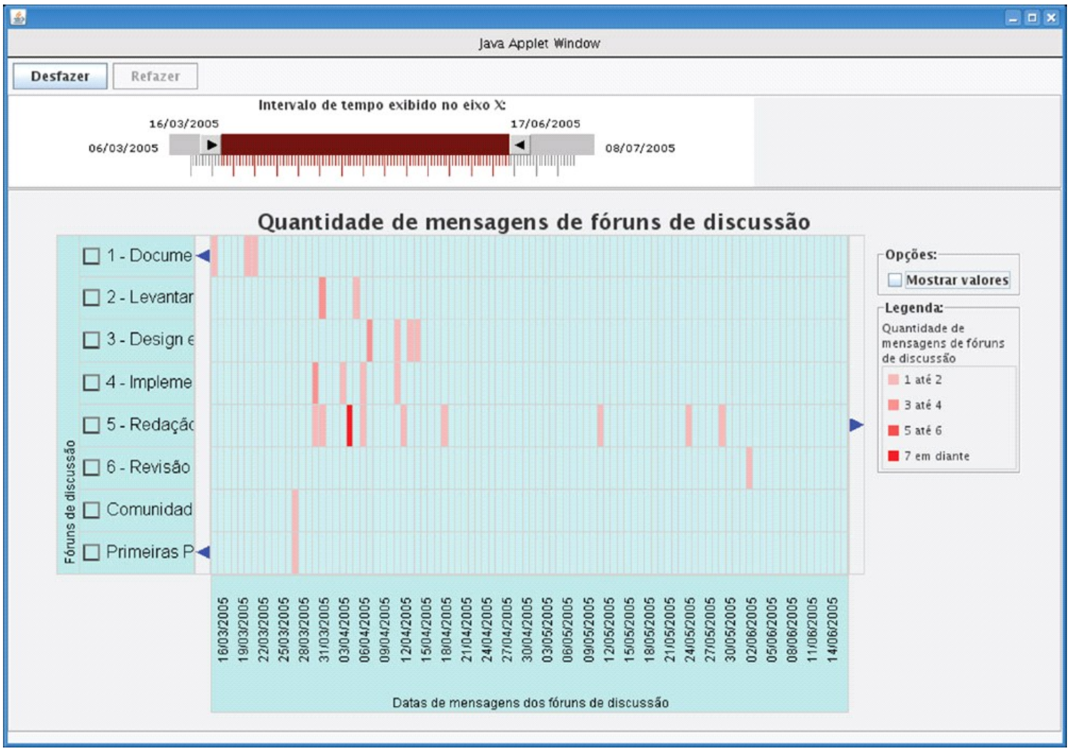

Figure 7: A heatmap showing results of a query

If user asks an alternative view for a dataset presented at a heatmap line or column, LMS Database Explorer defines a new visual structure for better presenting the selected data. Figure 8 presents this situation, in which a user selects another view for a heatmap line (a selected discussion forum). The visual structure used for presenting this view is a bar chart with the remaining attributes ("discussion forums messages date" and "discussion forums messages – quantity"). An available listbox enable user to change what discussion forum the presented data refers to.

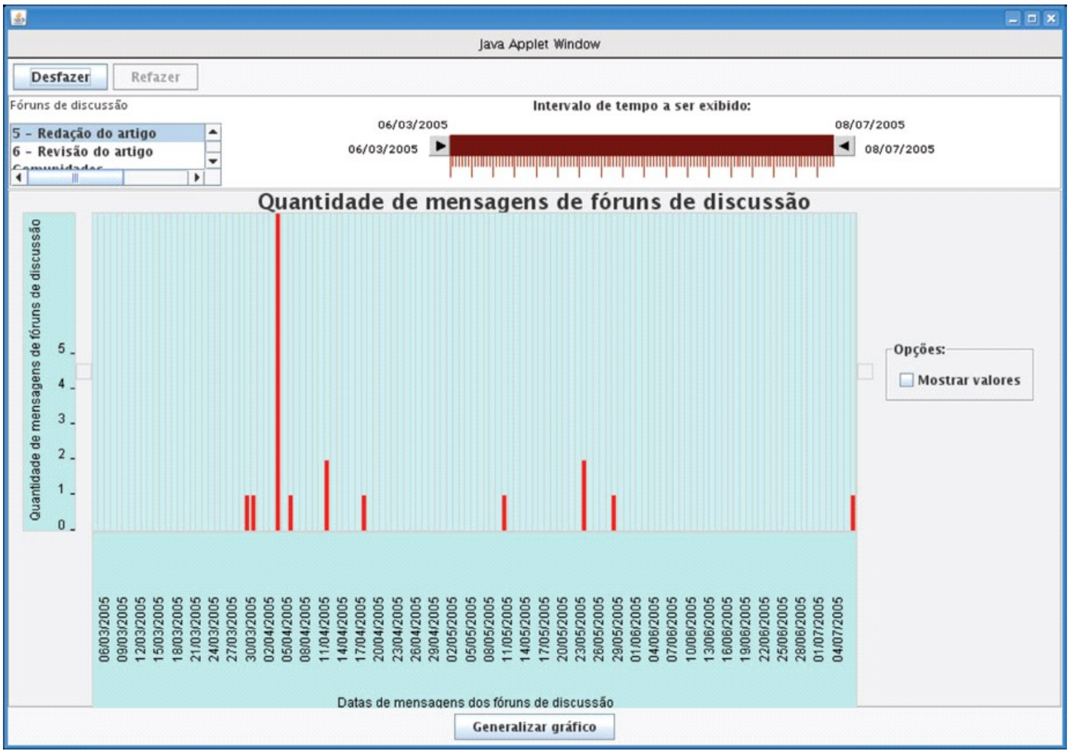

Figure 8: Bar chart showing data presented by the fifth line of previous heatmap

3.5 Prototype Evaluation

The implemented prototype was evaluated through a preliminary evaluation with three phases. In the first one, a set of LMS-related questions to be answered in the following phases was defined. In phase II, the author used the prototype for trying to answer the proposed questions. Phase III was a study case with users, in order to analyze their difficulty for using the prototype when answering the proposed questions. A detailed version of this evaluation is documented elsewhere (Silva, 2006).

Four users were involved in this evaluation; two of them are graduated in Computing and the other two in Education. All users have strong experience with e-learning courses offered through TelEduc, and provided valuable commentaries and suggestions about the software.

In phase I, these users and the author proposed questions to be answered by the prototype. Their experience with e-learning courses inspired them to come up with relevant questions in this context. A heterogeneous list of questions was proposed, with 5 questions proposed by the author, and 34 by the users. The following list presents some of these questions:

1. What participants are recently accessing the e-learning environment?
2. How intense has message sending at the Forum Discussion been by participant in the course?
3. How course participants have been talking to each other at the Chat tool?
4. How many (and which) students stopped to access the environment?
5. How many (and which) students have been using communication tools?
6. Which participant commented the production of which other participants?
7. What tools were more used by each user in a defined period?
8. What type of communication takes place more often: student-student or teacher-student? Is there equilibrium? (Analyze these types of communication in distinct periods of the course).

The heterogeneity of the proposed questions reinforces the necessity of a flexible tool like the proposed one, which tries to fit not only predefined questions about LMS, but also new questions created on demand.

In phase II, the author tried to answer the proposed questions. From the 34 proposed questions, 7 were not considered for the study because their answers would need either some data that are not stored in the LMS database, or data from functionalities that were incorporated to LMS after the prototype development. From the remaining 27 questions, 11 (40.7%) were completely answered, 9 (33.3%) were partially answered, and 7 (26.0%) were not answered.

Figure 9 presents an example of the completely answered question "How many (and which) students have been using communication tools?". In order to answer it, three database attributes were selected: "users' most recently date accesses", "tools accessed by users", and "users that accessed TelEduc, or one of its tools". For these selected attributes, the prototype generated a heatmap in which tools are presented at the Y axis, users at the X axis, and data accesses by color. Light green represents most recent accesses. Given this figure, the author could see the communication-related tools, its related light green cells and their respective students.

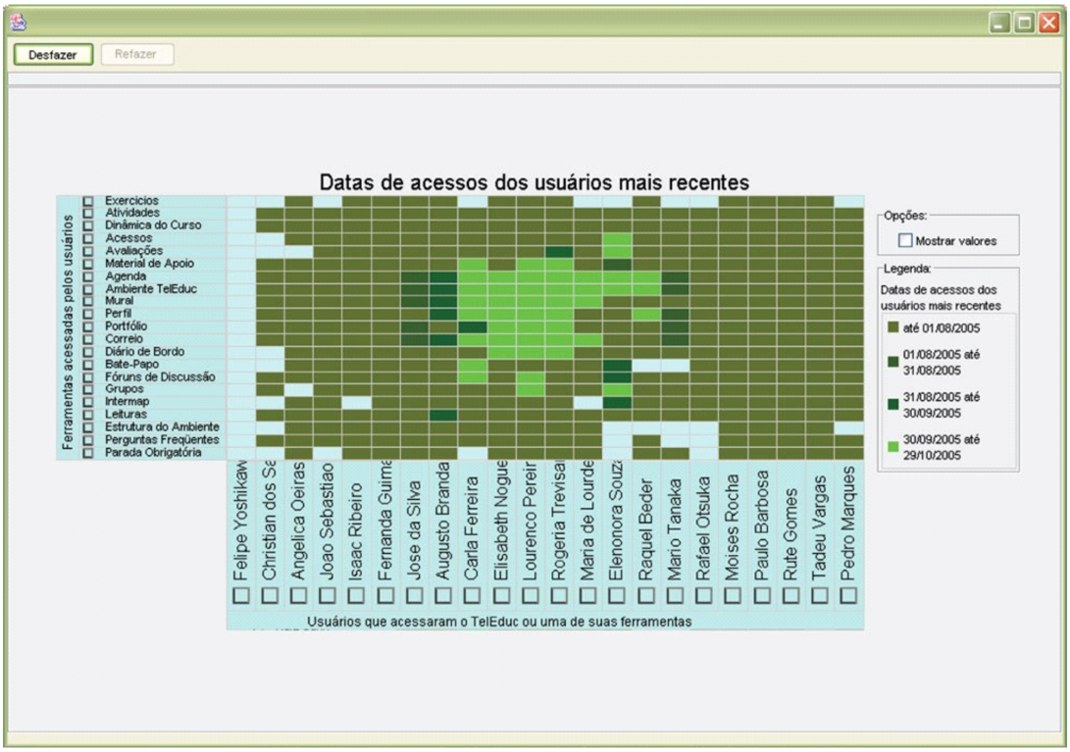

Figure 9: Most recent date of LMS tools accesses

Questions that were partially answered or not answered at all are related to absent functionalities, such as calculating percentages or mathematical expressions combining distinct attributes values. They were also related to the selection of attribute sets to which the prototype did not know how to generate a graphical representation. Some questions also revealed little system inconsistencies to be corrected.

The objectives of the third evaluation phase were 1) to analyze how difficult it was expressing queries through database attribute selection; 2) to analyze how difficult it was answering these queries, using the prototype-selected graphical representations; and 3) to obtain opinions about the prototype, which includes suggestions for its improvement. For this phase, user interactions with the prototype were recorded. These recordings enabled further analysis about which attributes were selected in order to answer each question, and about what output graph users expected to be presented by the prototype. It was also collected users' comments made along and after the prototype use.

All four users tried to answer the 5 questions elaborated by the author, except one user, which tried to answer only 4. From these 19 attempts, 16 (84%) generated a graph that helped users to answer these questions, either completely or partially.

Each user also tried to answer the questions he/she had created in phase one of this evaluation process. From these 26 questions, 14 (53.8%) were completely or partially answered (34.6% were completely answered and 19.2% were partially answered), and 12 (46.2%) were not answered.

As an example, Figure 10 shows a user answering the question "What was the intensity of Discussion Forums messages sent by participant in the course?". The user first selected two attributes – "Discussion Forums messages–quantity" and "Users who wrote forums messages". The prototype then shows a bar chart with the quantity of forum messages by message author.

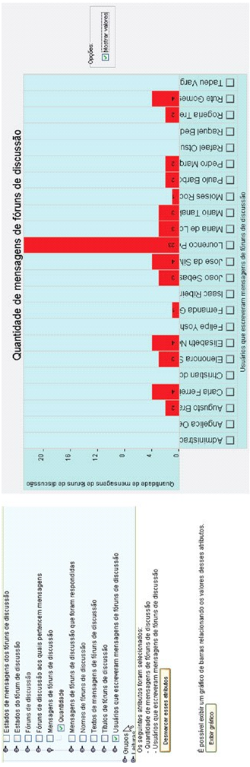

Figure 10: A user answers the question "What was the intensity of Discussion Forums message sent by course participant?" At the left, attribute selection screen At the right, graphical representation generated by the users attribute-based query This representation shows the number of messages sent by course participant into Discussion Forums tool

Users also filled in a post-use evaluation form. Their answers show that they had a good impression of the prototype. Those users agree that the prototype helps to analyze TelEduc data. According to the evaluation, users did not consider answering a query by the prototype a hard task. Their answer revealed that the presented visual representations were relevant and that the controls for data filtering and selection were very useful. Users' main suggestions in this evaluation were for taking care with attributes description (specially avoiding ambiguous attribute names), quantity of available attributes and prototype usability.

4 FUTURE RESEARCH DIRECTIONS

As presented in the Section 2, there is a multitude of LMS, including some open source systems such as TelEduc and Moodle. *Ad hoc* solutions have been developed for some of them in order to attend specific user needs of monitoring and/or assessing course participants. This chapter presents an approach which is, as far as I know, the only one that deals the problem of visually revealing data in a more general way.

In the other hand, the amount of software and APIs for data visualization is increasing. Nowadays programmers have good visualization APIs such as Prefuse, JIT, Data-Driven Documents and Google Chart Tools, which implement lots of interactive visual representations, mainly in Java and JavaScript. These tools are important accelerators for the prototyping and development of InfoVis solutions for e-learning and other domains. Besides, users without programming skills have available good visualization software such as the commercial tools Spotfire and Tableau, and the IBM Many Eyes site; all these software enable users to insert data and choose visual representations for representing it. Indeed, a signal that these software may be used for analyzing LMS data was just pointed out by the works of Hijón-Neira et al. (2008), Hijón-Neira & Velázquez-Iturbide (2008), and Gómez-Aguilar et al. (2008).

Therefore, it is worth to indicate some emerging and possible future directions for research and development in InfoVis applied to LMS:

- Providing mechanisms for exporting LMS data to visualization software (such as Tableau and Spotfire) is another way to enable teachers to visualize patterns, trends and outliers into their courses. It would be necessary to study both the usability of these software for typical LMS users, and the adequate data formats for each software. Optionally, users may connect visualization software directly to the LMS database, but there are ethical questions related to this option.
- Each of the presented researches developed tools for a single LMS. Establishing a common interface (such as an API or a web service) for obtaining data from LMS would improve software reuse and enable InfoVis tool exchange among distinct LMS. In this sense, SCORM (ADL, 2009) and AICC guidelines (AICC, 2004) are starting points for defining how to exchange learning resources; however, these works do not seem to deal with data such as accesses to LMS and interaction between course participants.
- E-learning and computer science researchers should direct efforts for evaluating the real impacts of adopting InfoVis tools into LMS scenario. There are lots of proposed visualizations but some works did only preliminary evaluation steps. One should provide more reliable answers for questions such as what are the most relevant visualizations for e-learning and how they actually affect this scenario.

5 CONCLUSION

Previous works from InfoVis related to LMS focused only on predefined subsets of LMS data, and hence they fit only a subset of users' information need about LMS stored data. This chapter presented a more general solution for querying LMS data and for representing the obtained data, fitting distinct LMS data analysis needs and enabling users to graphically analyze these data.

The proposed solution uses universal relation for enabling users to specify queries without knowing the database internal organization, which is a necessary feature for enabling the prototype use by people that have no database knowledge. Besides, analyzing attributes functional dependencies and categories is important for mapping what visual structure can express an attribute set.

The generality of the proposed solution resides on the fact that it is not connected to LMS concepts like "users", "discussion forum messages," or "tools", but to LMS databases and related metadata. Therefore, one may create extensions for connecting other LMS to the prototype.

Future works may focus on: using the prototype with other LMS (like TIDIA-Ae (Beder et al., 2005)) and with new TelEduc tools; enhancing query mechanisms and usability; internationalizing; expanding the tool for other knowledge domains; and implementing and using more visual structures.

This research was supported by CAPES and FAPESP (processes 2004/08233-5 and 2005/56629-8). Author thanks University of Campinas' General Coordination for its valuable reviews. He also thanks Prof. Dr. Heloisa Vieira da Rocha for all guidance provided for the author through his entire undergraduate and doctoral studies; Nucleus of Informatics Applied to Education (NIED) and Institute of Computing (IC) of University of Campinas, where this work was developed; and School of Technology, where this chapter was written.

REFERENCES

Advanced Distributed Learning–ADL. (2009). SCORM® 2004 4th edition-content aggregation model (CAM), version 1.1. *Department of Defense, United States of America.* Retrieved from http://www.adlnet.gov/capabilities/scorm/scorm-2004- 4th.

Ahlberg, C., & Shneiderman, B. (1994). Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, 313-317. New York: ACM Press.

AICC. (2004). CMI guidelines for interoperability, revision 4.0. *AICC CMI Subcommittee.* Retrieved from http://www.aicc.org/joomla/dev/index.php?option=com_ content&view=article&id=64&Itemid=28.

Beder, D. M., Otsuka, J. L., Silva, C. G., Silva, A. C., Talarico, N., et al., & Silva, J. C. A. (2005) The TIDIA-Ae portfolio tool: A case study of its development following a component-based layered architecture. In *Proceedings of 2nd Workshop TIDIA FAPESP*. São Paulo, Brazil: FAPESP Press.

Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Readings in information visualization: Using vision to think*. San Francisco: Morgan Kaufman Publishers.

Date, C. J. (1986). *An introduction to database systems, 1* (4th ed.). Boston: Addison-Wesley.

Elmasri, R., & Navathe, S. (2000). *Fundamentals of database systems* (3rd ed.). Boston: Addison Wesley.

Foroughi, R., & Taponecco, F. (2005). A visualization tool for student assessment and evaluation in online learning. In *Proceedings of IADIS International Conference on Applied Computing*, 68-73. IADIS Press.

France, L., Heraud, J.-M., Marty, J.-C., & Carron, T. (2005). Help through visualization to compare learners' activities to recommended learning scenarios. In *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies*, 476-480. Washington, DC: IEEE Press.

France, L., Heraud, J.-M., Marty, J.-C., Carron, T., & Heili, J. (2006). Monitoring virtual classroom: visualization techniques to observe student activities in an e-learning system. In *Proceedings of the Sixth International Conference on Advanced Learning Technologies*, 716-720. Washington, DC: IEEE Press.

Gómez-Aguilar, D. A., Conde-González, M. Á., Therón, R., & García-Peñalvo, F. J. (2011). Revealing the evolution of semantic content through visual analysis. In *Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies*, 450-454. Washington, DC: IEEE Press.

Gómez-Aguilar, D. A., Guerrero, C. S., Sanchez, R. T., Therón, R., & García-Peñalvo, F. J. (2010). Visual analytics to support e-learning. In Rosson (Ed.), Advances in Learning Processes (207-228). Rijeka, Croatia: InTech. doi:doi:10.5772/7932doi:10.5772/7932

Gómez-Aguilar, D. A., Therón, R., & García-Peñalvo, F. (2008). Understanding educational relationships in Moodle with ViMoodle. In *Proceedings of the Eighth IEEE International Conference on Advanced Learning Technologies*, 954-956. Washington, DC: IEEE Press.

Gómez-Aguilar, D. A., Therón, R., & García-Peñalvo, F. (2009). Semantic spiral timelines used as support for e-learning. *Journal of Universal Computer Science*, *15*(7), 1526–1545.

Hardless, C., & Nulden, U. (1999). Visualizing learning activities to support tutors. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)*. Pittsburgh, Pennsylvania: ACM Press.

Hijón-Neira, R. Velázquez-Iturbide, J. Á., Barn, B., & Oussena,S. (2008). A comparative study on the analysis of students interactions in e-learning. In *Proceedings of Eighth IEEE International Conference on Advanced Learning Technologies*, 20-22. Washington, DC: IEEE Press.

Hijón-Neira, R., & Velázquez-Iturbide, J. Á. (2008). How to improve assessment of learning and performance through interactive visualization. In *Proceedings of Eighth IEEE International Conference on Advanced Learning Technologies*, 472-476. Washington, DC: IEEE Press.

Huang, M. L., Nguyen, Q. V., & Hintz, T. (2005). Attributed graph visualization of collaborative workspaces. In *Proceedings of the Computer Graphics, Imaging and Vision: New Trends (CGIV)*, 155-161. Washington, DC: IEEE Press.

Klerkx, J., Duval, E., & Meire, M. (2004). Using information visualization for accessing learning object repositories. In *Proceedings of the Eighth IEEE International Conference on Information Visualisation (IV)*, 465-470. Washington, DC: IEEE Press.

Mackinlay, J. D. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, *5*(2), 110–141. doi:10.1145/22949.22950

Martín, M., Álvarez, A., Fernández-Castro, I., Reina, D., & Urretavizcaya, M. (2011) Experiences in visualizing the analysis of blended-learning interactions to support teachers. In *Proceedings of 11th IEEE International Conference on Advanced Learning Technologies*, 265-266. Washington, DC: IEEE Press.

Mazza, R. (2006). A graphical tool for monitoring the usage of modules in course management systems. In *Proceedings of Visual Information Expert Workshop*. Paris, VIEW Press.

Mazza, R., & Dimitrova, V. (2005). Generation of graphical representations of student tracking data in course management systems. In *Proceedings of the 9th IEEE International Conference on Information Visualisation*. London: IEEE Press.

Mazza, R., & Milani, C. (2005). Exploring usage analysis in learning systems: gaining insights from visualisations. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED)*, 65-72. Amsterdam: IOS Press.

Nguyen, Q. V., & Huang, M. L. (2005). EncCon: An approach to constructing interactive visualization of large hierarchical data. *Information Visualization*, *4*(1), 1–21. doi:10.1057/palgrave.ivs.9500087

Otsuka, J. L. (2006). *Multi-Agent Model to Formative Assessment Support at Learning*. (Doctoral thesis). Campinas, Brazil, University of Campinas.

Otsuka, J. L., & Rocha, H. V. (2007). A multi-agent formative assessment support model for learning management systems. In *Proceedings of the 7th IEEE International Conference on Advanced Learning Technologies*. Niigata, Japan: IEEE Press.

Rocha, H. V., Kropiwiec, D. D., Fukaya, S. M., Neto, J. C., Gayard, L. A., et al., & Ferreira, T. B. (2002). TelEduc project: Technology research and development for distance learning. In *Proceedings of IX Brazilian Association of Distance Education International Congress of Distance Learning.* Available at http://www.teleduc.org.br/artigos/premio_abed2002.pdf.

Romani, L. A. S. (2000). *InterMap: Tool for Visualizing Interaction in Distance Learning Environments on Web*. (Master thesis). Campinas, Brazil, University of Campinas.

Saltz, J. S., Hiltz, S. R., & Turoff, M. (2004). Student social graphs: Visualizing a student's online social network. In *Proceedings of ACM Conference on Computer Supported Collaborative Work (CSCW)*. Chicago: ACM Press.

Shneiderman, B. (1994). Dynamic queries for visual information seeking. *IEEE Software*, *11*(6), 70–77. doi:10.1109/52.329404

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, 336-343. Washington, DC: IEEE Press.

Silva, C. G. (2006). *Learning Management Systems' Database Exploration by Means of Information Visualization-Based Query Tools*. (Doctoral thesis). Campinas, Brazil, University of Campinas.

Silva, C. G., & Rocha, H. V. (2004). Contributions of information visualization for distance learning area. In *Proceedings of VI Brazilian Symposium on Human Factors in Computing Systems (IHC)*. Curitiba, Brazil: IHC Press.

Silva, C. G., & Rocha, H. V. (2007). Learning management systems' database exploration by means of information visualization-based query tools. In *Proceedings of Seventh International Conference on Advanced Learning Technologies (ICALT)*, 543-545. Washington, DC: IEEE Press.

Spence, R. (2001). *Information Visualization* (1st ed.). Boston: Addison-Wesley.

Teutsch, P., & Bourdet, J.-F. (2010). How to see training paths in learning management systems? In *Proceedings of 10th IEEE International Conference on Advanced Learning Technologies*, 349-351. Washington, DC: IEEE Press.

Tufte, E. R. (1990). *Envisioning information*. Chechire, CT: Graphics Press.

Tzoumakas, V., & Theodoulidis, B. (2005). Force based visualizations for instructor support. In *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies* (ICALT), 452-456. Washington, DC: IEEE Press.

Ullman, J. D. (1982). *Principles of database systems* (2nd ed.). Washington, DC: Computer Science Press.

Ware, C. (2004). *Information visualization: Perception for design*. San Francisco: Morgan-Kaufmann Publishers.

ADDITIONAL READING

Advanced Distributed Learning–ADL. (2009). *ADL web site.* Retrieved from http://www.adlnet.org/.

Affero. (2011). *AulaNet–Pioneer, innovator, and complete LMS*. Retrieved from http://www.eduweb.com.br/aulanet/.

AICC. (2004). *Aviation industry CBT committee (AICC) web site*. Retrieved from http://www.aicc.org.

Bakharia, A., & Dawson, S. (2011). Snapp: A bird's-eye view of temporal participant interaction. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge (LAK)*, 168-173. New York: ACM.

Berkeley Institute of Design. (2012). *Prefuse–Interactive information visualization toolkit.* Retrieved from http://prefuse.org.

Blackboard. (2012). *Blackboard site*. Retrieved from www.blackboard.com.

Bostock, M. (2012). D3.js–Data-driven documents. Web site. Retrieved from http://d3js.org.

Costagliola, G., Fuccella, V., Giordano, M., & Polese, G. (2009). Monitoring online tests through data visualization. *IEEE Transactions on Knowledge and Data Engineering*, *21*(6), 773–784. doi:10.1109/TKDE.2008.133

Dario, D. (2010). *Application of Techniques of Information Visualization to Support Formative Assessment in E-Learning Systems*. (Master dissertation). Campinas, Brazil, University of Campinas.

Dokeos Company. (2012). *Dokeos–Open source e-learning*. Retrieved from http://www.dokeos.com.

Donath, J., Karahalios, K., & Viégas, F. (1999). Visualizing conversation. *Journal of Computer-Mediated Communication 4*(4). Retrieved from http://dx.doi.org/10.1111/j.1083-6101.1999.tb00107.x.

Eskildsen, S., Rodil, K., & Rehm, M. (2012). Visualizing learner activities with a virtual learning environment: Experiences from an in situ test with primary school children. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT)*, 660-661. Washington, DC: IEEE Press.

García-Solórzano, D., Morán, J. A., Cobo, G., Monzo, C., Santamaría, E., & Melenchón, J. (2012). Educational monitoring tool based on faceted browsing and data portraits. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, 170-178. New York: ACM.

Gomez-Aguilar, D., Conde-Gonzalez, M., Theron, R., & Garcia-Peñalvo, F. (2011). Supporting moodle-based lesson through visual analysis. Human-computer interaction–INTERACT 2011. *Lecture Notes in Computer Science*, *6949*, 604–607. doi:10.1007/978-3-642-23768-3_93

Google. (2012). *Google chart tools*. Retrieved from https://developers.google.com/chart.

Hemphill, L., & Teasley, S. D. (2010). Overherd: Designing information visualizations to make sense of students' online discussions. In *Proceedings of the 9th International Conference of the Learning Sciences-Volume 2*, 302-303. New York: ACM Press.

Hijón-Neira, R., & Velázquez-Iturbide, J. Á. (2011). Merlin-Mo, An interactions analysis system for Moodle. In *Proceedings of the 16th*

Annual Joint Conference on Innovation and Technology in Computer Science Education, 340-340. New York: ACM Press.

IBM. (2012). *Many eyes*. Retrieved from http://www-958.ibm.com/software/data/cognos/manyeyes.

LRN. (2012). *LRN–Learn, research, network.* Retrieved from http://dotlrn.org.

LTSC. (2012). *IEEE learning technology standards committee (LTSC) web site.* Retrieved from http://www.ieeeltsc.org.

Mazza, R. (2004). *Using Information Visualisation to Facilitate Instructors in Web-based Distance Learning*. (Ph.D. Thesis). Lugano, Switzerland, University of Lugano.

Mazza, R. (2010). Visualization in educational environments. In Romero, Ventura, Pechenizkiy, & Backer (eds.). Handbook of Educational Data Mining (9-26). New York: CRC Press. doi:doi:10.1201/b10274-4doi:10.1201/b10274-4

Mazza, R., & Nidola, M. (2012). *GISMO–Graphical interactive student monitoring tool for moodle*. Retrieved from http://gismo.sourceforge.net.

Moodle. (2012). *Moodle–Project site*. Retrieved from www.moodle.org.

NIED/UNICAMP. (2012). *TelEduc–Project site*. Retrieved from www.teleduc.org.br.

Rabbany, R., Takaffoli, M., & Zaïane, O. R. (2011). Social network analysis and mining to support the assessment of online student participation. *ACM SIGKDD Explorations Newsletter*, *13*(2), 20–29. doi:10.1145/2207243.2207247

Rocha, H. V., Oeiras, J. Y., & Romani, L. A. S. (2001). Communication, visualization and social aspects involved on a virtual collaborative learning environment. *Journal of Three Dimensional Images*, *15*(3), 122–126.

Romani, L. A. S., & Rocha, H. V. (2000). Interaction map: Information visualization techniques in web -based distance education environments. *Technical Report (IC-00-17)*. Campinas, Brazil: Institute of Computing, University of Campinas. Retrieved from http://www.ic.unicamp.br/~reltech/2000/00-17.pdf.

Sakai Foundation. (2012). *Sakai Project–Collaboration and learning–For educators by educators*. Retrieved from http://www.sakaiproject.org.

SenchaLabs. (2011). *JavaScript InfoVis toolkit.* Retrieved from http://thejit.org.

Tableau Software. (2012). *Tableau Software web site*. Retrieved from http://www.tableausoftware.com.

TIBCO Software. (2012). *TIBCO spotfire–Business intelligence analytics software & data visualization*. Retrieved from http://spotfire.tibco.com.

Xiong, R., & Donath, J. (1999). PeopleGarden: Creating data portraits for users. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 37-44. New York: ACM Press.

Zhang, H., Almeroth, K., Knight, A., Bulger, M., & Mayer, R. (2007). Moodog: Tracking students' online learning activities. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 4415-4422. Chesapeake, VA: AACE Press.

KEY TERMS AND DEFINITIONS

CMI: Computer Managed Instruction. Another name for LMS.

CMS: Content Management System. Sometimes this name is used as a synonym of an entire LMS or of a subset of its tools.

DBMS: Database Management System.

Heatmap: A matrix whose cells are colored according to their values.

InfoVis: Information Visualization.

LMS: Learning Management System.

SCORM: Shareable Content Object Reference Model.

VLE: Virtual Learning Environment. Another name for LMS.

WebCT: An old LMS which was merged with Blackboard LMS.