

User Interface Design Guidelines for Rich Applications in the Context of Cultural Heritage Data

Patricia Martín-Rodilla

Institute of Heritage Sciences

Spanish National Research Council

Santiago de Compostela, Spain

patricia.martin-rodilla@incipit.csic.es

José Ignacio Panach

*Escola Tècnica Superior d'Enginyeria, Departament
d'Informàtica*

Universitat de València

Burjassot, Valencia, Spain

joigpana@uv.es

Oscar Pastor

Centro de Investigación en Métodos de Producción de Software

Universitat Politècnica de València

Valencia, Spain

opastor@pros.upv.es

Abstract—Advanced interaction techniques are necessary to explore the potential of large data-volume systems. In this context, rich internet application patterns were defined, but usually reduced to the development of social web applications. However, other types of applications, such as data-analysis applications, require also advanced interaction solutions to assist users in making decisions and data-analysis. This paper identifies a set of problems emerged in the interaction between humans and data-analysis applications. We propose a set of guidelines for rich applications as a solution for these problems. As illustrative example of a real data-analysis environment, the paper focuses on a case study in the cultural heritage domain, highlighting the existing interaction problems and how they can be solved through the design guidelines proposed. The set of design guidelines allows to specify interfaces abstractly, creating a repository to solve interaction problems. These guidelines aim to serve as a basis for a future identification of new rich applications design patterns.

Keywords—*Design Guidelines; Rich Application; Data Analysis; Cultural Heritage*

I. INTRODUCTION

In recent decades, the strategic importance of big data in decision-making has been growing exponentially, with emerging disciplines [1], professions [2] and techniques to assist humans in data management and interpretation of these large volumes of data. Decisions based on large amount of data are made in our day-to-day life in several areas, but especially in business (BPM -Business Process Management-) [3, 4], and data processing activities (qualitative research, statistical reasoning, etc.). In both areas, applications to assist users in data-analysis tasks and to facilitate the decision-making process are a key necessity. These applications are named data-analysis applications, and they require visualizing

large amount of data graphically. These applications use complex interfaces to avoid confusing elements such as large text lists or other elements with too much textual information unprocessed. Other characteristic of data-analysis applications is the high level of functionality required to assist end-users through the complete decision-making process.

The definition and design of interfaces for data-analysis tasks present some challenges for analysts, especially in modeling terms. We can classify these challenges into two main groups: analysts' skills and technical reasons. With regard to the first group, frequently, interface analyst and requirements engineer are roles played by different persons. This way, interface analyst ignores details of requirements and possible complex uses of the application. As a result, interfaces often implement lineal interaction mechanisms based on showing data, lacking of assistance for data-analysis tasks. The use of a wrong interaction mechanism can hinder tasks of analysis and reasoning based on data. The second group of challenges (technical reasons) appears due to analysis interfaces require the management of a large amount of data to facilitate tasks of analysis. This is a technical challenge in terms of screen space or interaction mechanisms. Furthermore, the design of interfaces becomes more complicated in technical terms because requirements are continuously evolving, more frequently in first stages of the development, due to the difficulty of extracting requirements related to cognitive and analysis tasks [5-7]. Users usually know which tasks they need to perform in each interface. However, users are not self-conscious about which interaction mechanism is better to perform each analysis task or which mechanism can improve the decision-making process.

Problems related to the definition of advanced interfaces have been tackled in depth in the software engineering field. Existing proposals focus on how to represent and reuse

interaction features in order to help analysts to create more adaptive interfaces to users' requirements. A common strategy to reach those goals is the use of interaction patterns.

In the context of interaction patterns we can find two types of patterns: (1) General patterns identified for graphical user interfaces commonly accepted; (2) RIA (Rich Internet Application), which represents advanced interaction features for Web applications. RIA patterns are usually focused on social network websites such as Facebook or similar collaborative environments, where there is a huge amount of interaction alternatives. However, there are few patterns for other types of less collaborative interactions with complex mechanisms to manage big data, such as applications involved in data-analysis to make decisions.

This paper aims to cover the existing gap of solutions to specify interfaces to assist the decision-making process through the use of design guidelines. These guidelines offer a solution to model complex interaction mechanisms of large-data volume systems. The identification of these design guidelines is based on the study of common problems in cultural heritage area, where understandable interface definitions is a challenge faced by interface analysts and requirements engineers. The target is to solve interaction problems identified in data-analysis systems such a way, these solutions can be reused in any other software development easily. For this reason, the guidelines are defined abstractly through a hierarchical structure. Our target in the long term is to transform these guidelines into RIA patterns. For this aim, we will study which guidelines are useful for several developers and are applied in several developments.

The use of design guidelines for specifying interfaces of data analysis applications offers a set of reusable solutions to solve problems. Using guidelines, we improve traceability from requirements elicitation to interface design. In addition, design guidelines encapsulate more complex technical decisions such as the distribution of space inside the screen or the most appropriate interaction mechanism depending on the situation.

The article is divided into the following sections: section 2 shows related work focused on interaction patterns and other design solutions for data analysis applications. Section 3 identifies modeling challenges in data-analysis applications through a case study. Section 4 defines a set of design guidelines to solve interaction problems of data-analysis applications. Section 5 applies the guidelines to the cultural heritage case study. Finally, section 6 describes conclusions and further works.

II. RELATED WORK

The increasing demand of systems to extract data and to help in the decision-making process stimulates the creation of data-visualization solutions to work with much information in different heterogeneous fields, for instance journal infographics [8] or communication and online courses [9]. Most of these solutions are ad-hoc visualizations that are created specifically on-demand to represent a single dataset (a collection of related data) or they are based on specific visualizations techniques, such as treemap [10]. These ad-hoc

solutions do not allow the reuse of interaction solutions. Nowadays, there are some studies about the definition, design and evaluation of specific visualization techniques, as well as the presentation based on data types or audience. Some of these approaches deal with abstract specifications of visualization techniques, originating abstract libraries for implementation [11-13]. The specification and implementation of code libraries is managed as a whole. This situation presents some shortcomings in the reuse of components due to the strong connection between the underlying conceptual model, which represents the interaction solution adopted, and a specific implementation of the solution.

There are also new efforts to create specification languages to facilitate the definition of information visualization interfaces. One example is the language created by IBM [14] to define an abstract specification of interfaces and the reuse of interaction solutions. In our opinion, this solution demands specific analyst's modeling skills, which reduces its applicability. In most cases, the learning process of these skills has a hard learning curve. Specification languages are an integral solution for large teams or companies [15] but they are difficult to implement in other contexts.

Both approaches, code libraries and specification languages are applied to data-analysis applications. This way, we find data-analysis applications with ad-hoc models based on a domain [16] and with models based on a specific information visualization technique, as explained at the beginning of this section. We also find data-analysis applications with code libraries and languages for interface specification, trying to solve some of the ad-hoc problems. However, these approaches increase the need of modeling skills and couple the solutions with a specific implementation.

A solution based on abstract models, easily understandable by analysts and end-users, and independent of implementation is needed. The concept of pattern is a solution for those challenges.

The use of patterns consists in the repetition of a solution applied to similar problems independently of domain and of analyst's previous skills. There are several works to model interaction based on patterns, from element-based to object-oriented or tasks-oriented approaches. OO-Method [17], for instance, is an object-oriented design tool to develop information systems based on the Model-Driven Development paradigm. OO-Method includes conceptual primitives to associate interaction elements with objects of the domain. These interaction elements are defined using a set of interaction patterns expressed in a pattern language called Just-UI [18]. There are other modeling solutions based on UML [19] such as WISDOM (Whitewater Interactive System Development with object Models)[20], a method of software engineering expertise for the construction and maintenance of interactive applications for SMEs.

Existing approaches show the advantages of using patterns to develop business applications, but they also present some limitations. Patterns in OO-Method are limited to the scope of form-based applications. WISDOM approach is limited to the scope of Small Software Developing companies (SSDs), showing an interaction modeling solution more suitable for

working with small data than for supporting a decision-making process. OO-Method and WISDOM approaches have a strong dependency between models to represent the interaction and models to represent other features such as functionality or persistency. This dependency hinders to apply the interaction modeling to data-analysis applications, since functional and persistency models of OO-Method and WISDOM are not prepared to represent data-analysis systems.

With regard to web environments, Valverde [21] performs a complete study of interface specifications. That study highlights the extension of UsiXML [22] to support the generation of RIA interfaces. The application of UsiXML to data-analysis interface specification presents similar dependency problems and requires much modeling skills for analysts.

Finally, RIA patterns are reusable solutions for common problems in interface specifications [23] that are focused on collaborative environments or social networks [24]. For instance, we can find RIA patterns to solve web integration and searches (such as mechanisms to manage active or inactive search results).

According to the literature, interaction specifications based on patterns has a set of advantages: an integral treatment of the interaction mechanisms; avoiding ad-hoc specification; there is not excessive connection between implementation and specification, avoiding the hard learning curve of specification languages. The use of patterns fits in our purpose to offer a solution to model interaction mechanisms in data-analysis applications, mitigating identified problems. Note that despite existing attempts in the community, there is not a standard to define patterns. This situation results in a set of multiple patterns for different purposes: Web environments, social networks, etc. Our approach is a first step to build patterns for data-analysis systems. This paper identifies a set of design guidelines useful for the specification of interfaces in the area of data-analysis systems. In next steps of our investigation, we are planning to define new RIA patterns through the study of these design guidelines during the software development process with several developers.

This paper tackles the need of solutions to define interaction for data-analysis applications. We have analyzed eight data-analysis applications in order to identify challenges in the field and define a set of design guidelines to design that kind of applications. The main objective of our work is the definition of a catalogue of design guidelines to expand the scope of RIA solutions to data-analysis systems.

III. CHALLENGES OF DATA ANALYSIS SYSTEMS

This section analyses current challenges to specify data-analysis applications, more specifically we focus on the interaction with end-users. All these challenges have been obtained after analyzing eight data-analysis systems in the cultural heritage domain. The study is based on previous[25] interviews with end-users and developers of those systems. Studied systems share characteristics in terms of data management [26-28], processes [29] or visualization [30] in order to assist in the data-analysis and knowledge generation.

Identified challenges can be summarized in two items: analyst's skills and technology.

Analyst's skills are necessary for designing interfaces that help end-users to take decisions with big data. However, analysts of data-analysis applications have usually a huge background in specifying persistency and behavior but they lack of expertise in interaction design. A challenge is to help analysts specify interfaces even though they are not experts in this field. Regarding technological aspects, we have detected five modeling challenges in the definition of data-analysis environments:

- Challenge 1: Need for modeling large data volume presentation dealing with screen space limitations.
- Challenge 2: Need for modeling a big dynamism and more interaction options to visualize dynamically data in function of the user's reasoning in each moment.
- Challenge 3: Need for modeling the management of importance levels: A set of data can play different roles in function of the tasks performed by the user in each moment.
- Challenge 4: Need for modeling the use of interaction elements (colour, size, etc.) as interaction resources to improve the data-analysis performed by the user.
- Challenge 5: Need for modeling strongly dependencies and data characteristics, especially in time-oriented data and geographical aspects.

All these technological needs are frequent in data-analysis systems to visualize large amount of data and to improve user experience and specially, in traditional analytic domains related to end-users' strategic decisions. Business analysis is a good example and cultural heritage context is not an exception of these technological needs. The following case study shows some of the data-analysis challenges in the domain of cultural heritage.

A. Case study

A common need in cultural heritage areas is the analysis of material evidences found in an intervention or archaeological excavation, producing datasets with information about the characteristics of evidences. We take as case study one real dataset from the archaeological intervention in Alto do Castro [31], an Iron Age barrow situated in the northwest of Spain, with information regarding ceramic pots found at the archaeological site. The barrow has been excavated and documented by Incipit research team, as well as some management information needs have been identified by them [25]. Previous studies about user needs' in the area [25], in order to capture needs and requirements, confirm the existence of the challenges explained before. The reason to choose this case study is because it gathers all the most common challenges of data-analysis systems.

Researchers of the archaeological excavation aim to analyze the information available considering different parameters related to ceramic (decoration, shapes, etc.). In this

context, the archaeological team of Alto do Castro has detected the need of building a software system to manage, visualize and assist the archaeologist in analysis tasks. This system is the heart of the archaeological research process: the archaeologists share their data; analyze data to extract conclusions about the excavation; search for specific data; visualize data to make decision. The analyst needs a solution to specify interaction mechanisms in data-analysis. General technological challenges previously described for any data-analysis system are instantiated for our specific case study into the following problems, derived from interaction requirements. Problems are specified in the same order as challenges were defined previously:

- PROBLEM A: Users need a general view of information contained in a dataset with ceramic pots and their contour. This information clarifies the content and helps users decide the analysis strategy. This view must show all ceramic pots and fragments with information about size, border lines, etc. Depending on the existing interface (landscape or vertical), end-users would like to choose between showing the available information about each contour type in one row or in one column.
- PROBLEM B: Users need to group potteries by one criterion, and change this grouping criterion continuously in order to detect hidden subsets (dynamic groups of related data following a criterion) or relationships among them.
- PROBLEM C: Users need to focus the level of importance on one subset, but also maintaining contextual information, such as the proportion between this subset and the total of data available. For instance, the user highlights the group of decorated potteries but wants to maintain the information about the percentage of non-decorated potteries (contextual data).

- PROBLEM D: Users need to identify easily subsets displayed in the interface (for example, using same colour or same size). This way, end-users can recognize similarities or differences among data quickly.
- PROBLEM E: Users need to visualize one strongly temporal attribute of the pottery information: the conservation state. This attribute can change its value (restored, deteriorated, destroyed) depending on preservation activities of each pot.

Apart from all the technological challenges, we have also the challenge of analysts' skills. The analyst must face up to how to represent all these interaction requirements abstractly. There are well known models to represent data (UML class diagram) and behavior (UML activity diagram). However, there is not a notation widely used to represent interaction features, even though there are some standards, such as Interaction Flow Modeling Language [32]. As a mechanism to help analyst in the interface specification and to reuse interface solutions, we propose the definition of Rich Internet Application (RIA) patterns to face with all the interaction problems of data-analysis systems. The definition of the set of guidelines based on RIA specification help us to offer a description of a problem and a solution [33] in an abstract and structured way. Next section describes the guidelines identified to solve all the technological challenges previously described.

IV. DESIGN GUIDELINES DEFINITION

This section identifies a set of design guidelines and explains them in detail according to the most common specification structure of RIA solutions [34]: a title, a problem, a context of application and a solution. In addition, the specification includes a motivation to explain why we should use each guideline. These guidelines aim to solve challenges identified previously and to build a repository of

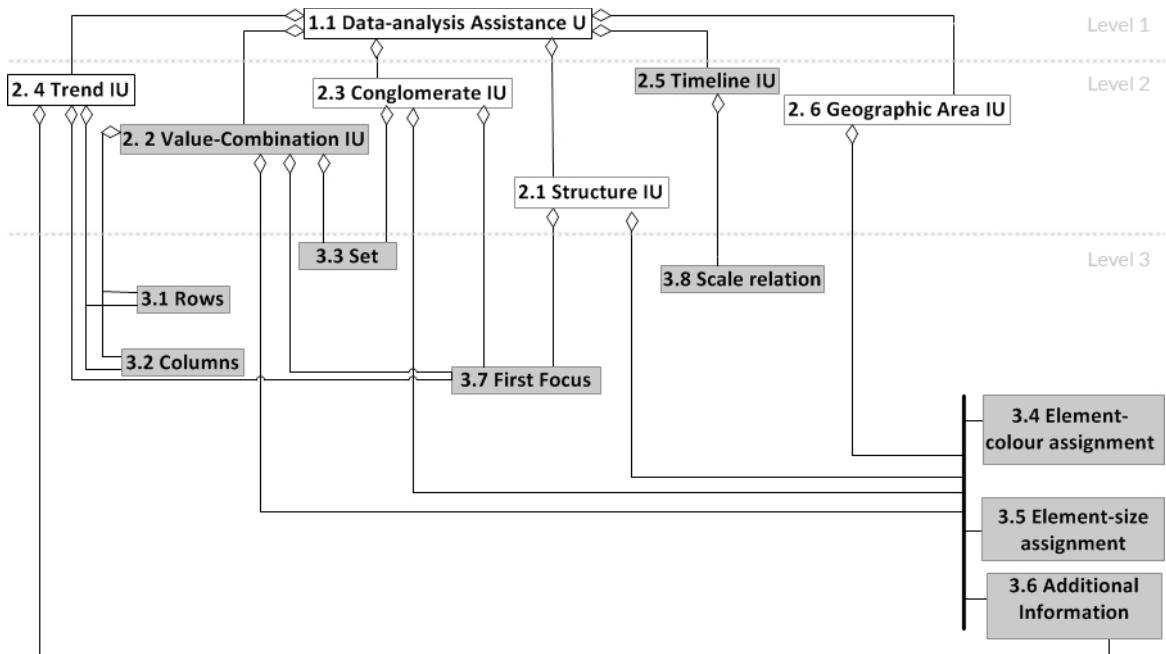


Fig. 1. Overview of defined guidelines per levels

solutions for the development of data-analysis systems.

Existing research in interaction and design modeling[35] has identified the need of working with different abstraction levels to define abstract solutions. The classification of solutions from highest to lowest level of abstraction allows to encapsulate behavior in levels and to reuse single guidelines or patterns in more complex solutions specification [35]. Following this approach, we define three levels of guidelines:

- Level 1: Data-analysis Assistance Unit. This is composed by one single unit: the Data-analysis assistance unit. This artifact encapsulates interaction units available to assist users in performing data-analysis tasks identified. This guideline is an abstract representation of a menu to navigate through interfaces. Note that since we have only one pattern in level 1, we do not describe more elements of this level in next sections.
- Level 2: Interaction Units. This identifies Interaction Units (IU). An IU is an abstract representation of an interface that will be used by end-users to carry out analysis tasks. Each IU can be seen like a container that gathers a set of individual guidelines identified in the third level.
- Level 3: Individual Guidelines. This identifies Individual Guidelines inside IUs. Each individual guideline can be used in several IUs. An Individual Guideline is an abstract representation of a widget in the interface with a specific behavior.

All the guidelines that compose the suite showed in Fig.1 are explained in detail in next sections, explaining the solution provided in each case to solve the related challenges. A guideline represented with a grey background highlights the guideline used in our case study.

A. Level 2

1) Structure IU.

a) *Problem:* Users need a view of the structure used in a dataset, independently of the information displayed.

b) *Context of application:* The user demands to visualize the structure of the information in function of classes, attributes and associations.

c) *Solution:* Organize the structure of the information contained in the dataset following object-oriented criteria (classes, attributes and associations) but also hiding technical aspects to the user (data types, database technical specifications). This solution deals with Challenge 1.

d) *Motivation:* The analysis based on the structure of information is a common practice in data-analysis applications. For instance, the user can see attributes of different classes in order to decide whether or not the comparison of these classes is interesting and what criteria can be used for the comparison.

2) Value-Combination IU.

a) *Problem:* Users need features to search and evaluate data in function of the value of class attributes.

b) *Context of application:* The user demands to search behaviors to classify data in function of the value of attributes.

c) *Solution:* Organize the information contained in a dataset, allowing to change target classes for analysis and to show the information of these classes. This solution deals with Challenges 1, 2 and 4.

d) *Motivation:* The analysis based on values of attributes allows the user to reason out statistical characteristics of the dataset. For instance, the user can evaluate attributes to reason out averages, percentages, etc.

3) Conglomerate IU.

a) *Problem:* Users need features to classify data contained in a dataset.

b) *Context of application:* The user demands a dynamic mechanism to classify a dataset by criteria.

c) *Solution:* Organize the information allowing the dynamic classification of datasets, improving user-friendliness. This solution deals with Challenges 2 and 4.

d) *Motivation:* The analysis based on classifications helps the user to understand the main entities presented in a dataset and data-deviations. For instance, the user can detect deviations regarding outliers or atypical values.

4) Trend IU.

a) *Problem:* Users need features to deal with the composition of elements within a dataset.

b) *Context of application:* The user demands to visualize a dataset depending on the compositions of elements.

c) *Solution:* Organize the information in function of the number of elements that are part of a specific criterion and changes on their composition. This solution deals with Challenges 3 and 4.

d) *Motivation:* The analysis based on composition allows the user to reason out trends in a dataset. For instance, the user can see how many entities belong to a specific class during different moments.

5) Timeline IU.

a) *Problem:* Users need features related to time-oriented analysis, especially in the analysis of time-oriented attributes – attributes with different values through time-.

b) *Context of application:* The user demands to see changes in values of attributes through time and to relate them to other data.

c) *Solution:* Organize the information to select, display and change time-oriented attributes in a time-oriented interface. This solution deals with Challenge 5.

d) *Motivation:* The analysis based on time-oriented data allows the user to reason out temporal dependencies among

data. For instance, the user can see how the values of two attributes change through time to analyze possible influences between them.

6) *Geographic Area IU.*

a) *Problem:* Users need features related to geographical analysis.

b) *Context of application:* The user demands the visualization of data based on geographical attributes.

c) *Solution:* Organize the information contained in the dataset in function of existing geographic information in a geographical-oriented interface. This solution deals with Challenge 5.

d) *Motivation:* The analysis based on geographic information is a common practice in data-analysis applications. For instance, users can see data related to their source locations in order to detect geographical relationships.

All Interaction Units are represented in Fig. 1 ordered by numbers from 2.1 to 2.6, belonging to level 2.

B. Level 3

1) *Rows aggregation guideline.*

a) *Problem:* Users need a general view of information managed on interfaces with landscape orientation. In addition, there is a need of consistency amongst other visualization elements of the screen.

b) *Context of application:* The user demands to visualize information in function of: a characteristic in a landscape interface –avoiding scroll behaviors-, a characteristic based on categories or based on intervals of values.

c) *Solution:* Organize the information in rows, creating divisions per row in the interface. This allows an analysis from left to right with a good distinction between the categories or intervals of values. This solution deals with Challenge 1.

d) *Motivation:* The analysis based on categories or intervals of values is a common practice in data-analysis applications. Rows aggregation can help distribute analysis results in landscape orientation intuitively.

2) *Columns aggregation guideline.*

e) *Problem:* Users need a general view of information managed on interfaces with vertical orientation. In addition, there is a need of consistency amongst other visualization elements of the screen.

f) *Context of application:* The user demands to visualize information in function of: a characteristic in a vertical interface –avoiding scroll behaviors-, a characteristic based on categories or based on intervals of values.

g) *Solution:* Organize the information in columns, creating divisions per column in the interface. This allows an analysis from up to bottom with a good distinction between the categories or intervals of values. This solution deals with Challenge 1.

h) *Motivation:* The analysis based on categories or intervals of values is a common practice in data-analysis applications. Columns aggregation can help distribute analysis results in vertical orientation intuitively.

3) *Set guideline.*

a) *Problem:* Users need to visualize a set of data aggregated by a criterion.

b) *Context of application:* The user demands to visualize information aggregated by a criterion, which is generally an attribute of a class. The result of this pattern is a set of groups that classify a dataset.

c) *Solution:* Organize the information creating one element in the interface (sphere, bar, area, etc.) for each resultant group. This solution deals with Challenge 2. This pattern offers a mechanism to choose the aggregator attribute.

d) *Motivation:* The analysis based on categories or intervals of values is a common practice in data-analysis applications. Set pattern allows an easier analysis, focusing on specific groups of a dataset.

4) *Additional information guideline.*

a) *Problem:* Users need to query information about some specific elements in an interface with much information.

b) *Context of application:* The application of this pattern is to show contextual information avoiding popup windows or changes in the interface.

c) *Solution:* Organize the information showing concrete information in a little overlaying element responding to a user action (mouse contact, click, etc.). This solution deals with Challenge 3.

d) *Motivation:* The analysis to detect data within range and outliers is a common practice in data-analysis applications. If the user finds an unusual behavior of one sample, the system should offer additional information with a quick and non-intrusive mechanism. This way, the user can infer possible causes.

5) *First Focus guideline.*

a) *Problem:* Users need to filter visual noise in a complex data interface.

b) *Context of application:* The user demands to visualize several focuses or parts of some information showed in an interface with a high information density.

c) *Solution:* Organize the information using shadows or a disable mechanism to focus on the most important information for the user. This solution deals with Challenge 3.

d) *Motivation:* Interfaces to analyze several data involve having two or three focus of attention at the same time to analyze related data. This pattern offers a solution to cover the analysis needs without changes of interface.

6) *Colour assignment guideline.*

a) *Problem:* Users need to give semantic meaning to the relation between an interface element and a colour.

b) *Context of application:* The user demands to easily identify groups or elements in interfaces with much information.

c) *Solution:* Organize the information offering the user a mechanism to choose a colour to represent a set of data through all interfaces. In function of a criterion specified by the user, the interface colours a visual element (sphere, bar, area, etc.) that represents a set of data. This solution deals with Challenge 4.

d) *Motivation:* Colour selection provides an effective mechanism for non-intrusive grouping. Colour assignment pattern helps maintain a colour-logic through all interfaces.

7) Size assignment guideline.

a) *Problem:* Users need to give semantic meaning to the relation between an interface element and its size.

b) *Context of application:* The same as Colour assignment pattern.

c) *Solution:* Organize the information offering the user a mechanism to choose the role of the element size in the representation of information through all interfaces. In function of the criterion chosen, the interface changes the visual element size (sphere, bar, area, etc.). This solution deals with Challenge 4.

d) *Motivation:* The same as Colour assignment but referring to maintain a size-logic consistency through all interfaces.

8) Scale relation guideline.

e) *Problem:* Users need to show different values of attributes that can change through time.

f) *Context of application:* The user demands to visualize time-oriented data, with phases and events related to a sample in a temporary context. In this case, the user needs two separate timelines to represent different facets of the phases or events associated to the data.

g) *Solution:* Organize the information offering a mechanism to relate two different timelines in the same interface. This solution deals with Challenge 5.

h) *Motivation:* There is no solution to show data from two timelines at the same timeline. Scale relation pattern represents two separate timelines in the same interface showing relations between them: overlapping between phases, correspondence between events, etc.

All Guidelines are represented in Fig. 1 ordered by numbers from 3.1 to 3.8, belonging to level 3.

V. GUIDELINES IN ACTION: VISUALIZING ARCHAEOLOGICAL INFORMATION

We have evaluated our proposal through a case study. Note that there are several ways of performing an evaluation [36]: from a *validation* before the approach has been transferred to practice until the *evaluation* in a real context with real users. This section deals with a validation of our approach before its implementation in a real system. This type of validation suffers from predictions and some degree of uncertainty. However, it is the more suitable for an emerging approach where we aim to check it before spending any effort in their implementation in a practical setting. Next, we describe how guidelines previously

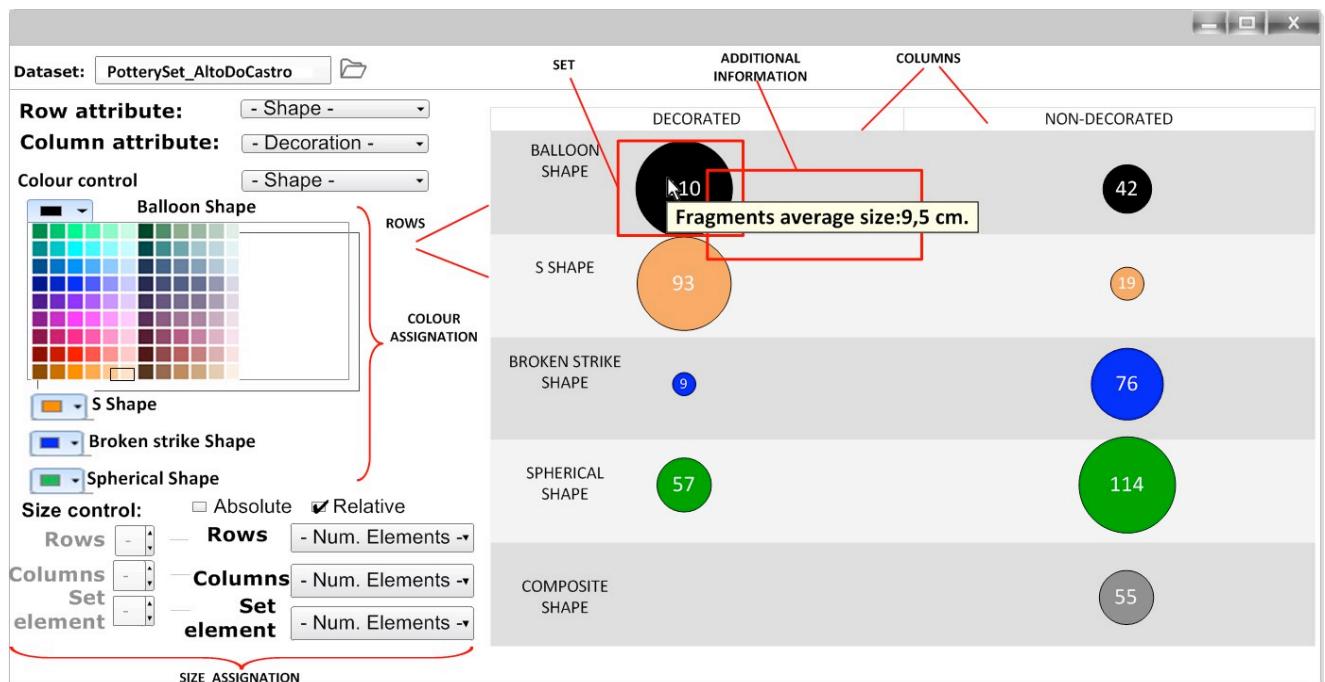


Fig. 2. Value combination guideline implementation with some guidelines of level 3 referenced to support problems A, B, C, D

defined for any data-analysis system can be used in the specification of our case study. As case study, we continue with the development of a system for visualizing archaeological information, describing how problems identified in Section 3 are solved with guidelines proposed.

Regarding Problem A, we can apply the guidelines of level 3, Rows aggregation or Columns aggregation –depending on the interface position – to organize a large amount of data in the screen. In our example, we aim to display all information related to pots. In Fig. 2, we are applying both guidelines at the same time: Rows aggregation shows information about the shape of pots and Columns aggregation shows information about the decoration of pots.

Regarding Problem B, we can apply the guideline of level 3 Set to group ceramic pots by decoration and shape. Moreover, Set shows the number of ceramic pots in each group. Thus, the user can perceive each group as a whole and compare them to each other.

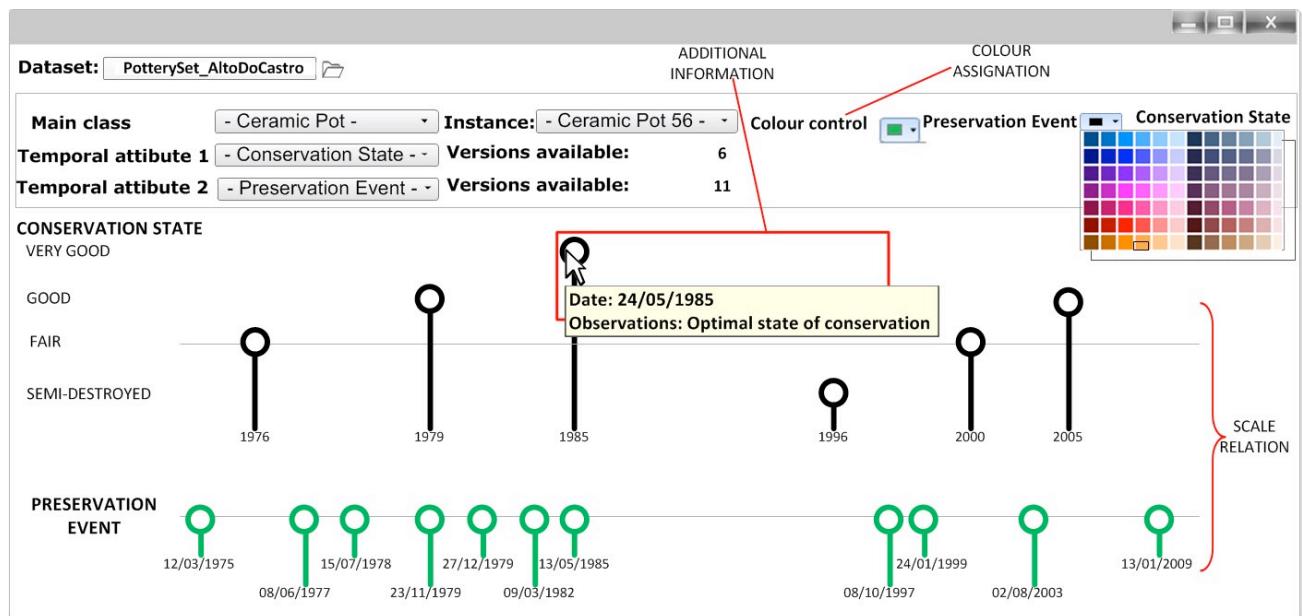


Fig. 3. Timeline guideline implementation and level 3 guidelines referenced to support problem E

To solve Problem C, we can apply the guideline of level 3 Additional Information: Fig. 2 shows the average size of the pottery's fragments in each group as a tooltip when the user moves the mouse over a specific group. This information is relevant for a deeper data-analysis: groups of ceramics that contain fragments with a small aver-age size indicate that the pots were quite broken and belonged to other complete pots.

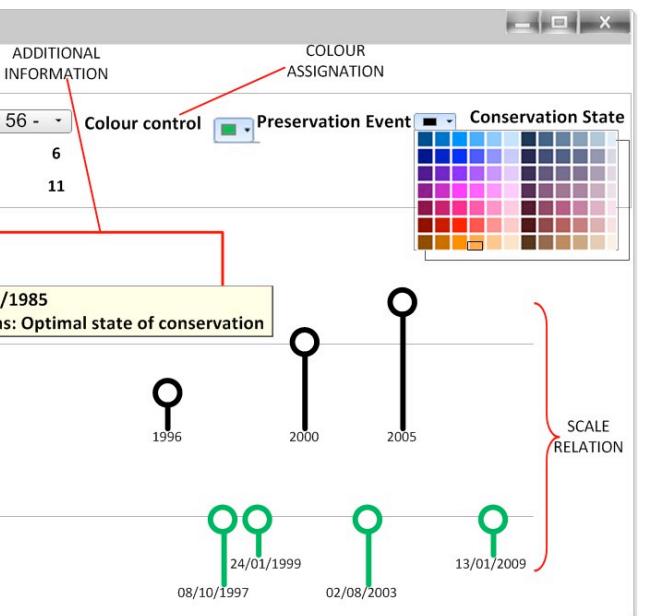
Regarding Problem C, we also apply the guideline of level 3 First Focus: if the user clicks on a ceramic group, the interface emphasizes this group, leaving the rest of the interface elements in the background. In Fig. 2, if the user would like to focus on the group of decorated pots with balloon shape, the system will disable with a grey background all the elements of the interface except for the black sphere. Thus, the

interface shows the same set of information, but with an additional emphasis on a particular area.

Regarding Problem D, we can apply guideline of level 3 Colour assignment to apply different colours depending on the shape. In addition, we apply guideline of level 3 Size assignment: the size of the icons reflects the number of elements in proportion to the full dataset. As we can see in Fig. 2, these guidelines help indicate similarities and differences among created groups.

As we can see, Problems A, B, C and D are related to in-depth analysis of ceramic pots data based on the values of their attributes; mainly shape and decoration. The Value-Combination IU (level 2) can be applied to solve these problems, encapsulating level 3 guidelines with a grey background in Fig.1.

Regarding Problem E, we can apply the guideline of level 2 Timeline Interaction Unit to support analysis of time-oriented attributes of ceramic pots, in our case, “Conservation State”



and “Preservation Event”. Within this guideline, we apply the guideline of level 3 Scale relation. As we can see in Fig. 3, the different values for the attribute “Conservation State” of one specific ceramic pot are showed in a temporal scale. In the temporal scale we can see the different values through time. For instance, in 1985 the conservation state of the pot was very good, whereas in 1996 was semi-destroyed. The different values of “Preservation Event” correspond to activities of preservation carried out in the analyzed ceramic pot. In this way, the user can know which preservation activities were held in the ceramic pot and on which date, analyzing their potential impact on the state of the pot through time, including the current state of the ceramic pot.

As we can see in Fig. 2 and Fig. 3, guidelines of level 3 can be reused through different IUs of level 2. This allows us to

share the same solution in several IUs. For example, Fig. 2 and Fig. 3 share the guidelines Additional Information and Colour Assignment in different contexts. Both Fig. 2 and Fig. 3 are accessible through a menu specified with the pattern of level 1 Data-analysis Assistance Unit.

In conclusion, the hierarchy of guidelines offers a catalogue of interaction solutions depending on the granularity of the problem: concrete problems are specified in level 3; more general problems related to deep analysis tasks are defined in level 2; and the data-assistance unit is defined in level 1, which encapsulates all the behaviors of the system.

VI. CONCLUSIONS AND FUTURE WORK

Applications focused on assisting users in the process of taking decision based on data require interaction solutions for data-analysis tasks. The definition of interfaces that implement these data-analysis tasks is not trivial, since they involve much functionality and interactive features. We have identified a set of guidelines to specify the definition of interfaces for data-analysis systems to reduce technological challenges and to help the analyst in interface design. Identified guidelines are domain-independent and they provide a design tool to improve end-user experience and to increase the level of assistance in data-analysis tasks. These guidelines have been extracted from a study based on interviews with end-users and developers of eight data-analysis systems in the cultural heritage domain.

This research is an initial point in the study of new interaction mechanisms in rich application that manage large amount of data. Identified guidelines emerge from end-users' needs in the perspective of data-analysis, but most of them can be useful for other related purposes (data reporting, control tasks, etc.).

The classification of guidelines in levels allows us to manage the abstraction and to combine simple guidelines to model more complex interaction mechanisms. All the possible combinations result in a high variety of possible interfaces to specify.

The use of a well-known specification structure based on RIA patterns to define the guidelines facilitates design tasks without constraints in implementation, as well as the creation of a repository with interaction solutions to reuse in different development projects, independent of platform and programming language. The developer decides the implementation for each solution during next steps of the software development. Note that ensuring the appropriate solution in implementation terms is a developer's responsibility. The repository only offers all the available solutions to interact with data-analysis systems.

Our future work focuses on four different aspects. First, we are interested in the deep evaluation of these design guidelines with a significant amount of users, in order to detect new RIA patterns. Patterns can only be defined when solutions of design guidelines are used by several developers in the development of several systems.

Second, we plan to define a set of guides to show which combinations of patterns are the most suitable to improve usability. Recommendations of patterns combination may depend on the type of user, context and tasks to optimize

effectiveness, efficiency and satisfaction according to ISO 25010.

Third, we are interested in the connection between defined guidelines and the cognitive process that the user is performing in data-analysis tasks. The case study used in this paper illustrates how existing problems in the domain can be solved through design guidelines. Our goal is to go one step further and identify the connection between the application of guidelines as a design solution and the improvement in the cognitive process that supports the guideline application. For instance, if we have several guidelines to represent a comparative process, which one presents better results in these comparative tasks?

Finally, we continue working in the detection of new needs of interaction in analysis tasks with large amount of data. The goal is to identify new design solutions that allow us to solve existing problems in the human analysis of big data, independently of domain.

ACKNOWLEDGMENT

This work has been developed with the support of MICINN (PROS-Req TIN2010-19130-C02-02), UV (UV-INV-PRECOMP13-115032), and co-financed with ERDF.

This work has been partially funded by Spanish National Research Council through contract for the program "Junta de Ampliación de Estudios". (B.O.E. 146, 20/06/2011).

REFERENCES

- [1] T. H. Davenport, et al., "Competing on talent analytics," *Harvard Business Review*, vol. 10, pp. 52-58, 2010.
- [2] D. Keim, et al., *Visual analytics: Definition, process, and challenges*: Springer, 2008.
- [3] T. H. Davenport, *Process innovation: reengineering work through information technology*: Harvard Business Press, 1993.
- [4] H. Smith and P. Fingar, *Business process management: the third wave* vol. 1: Meghan-Kiffer Press Tampa, 2003.
- [5] M. W. Van Someren, et al., *The think aloud method: A practical guide to modelling cognitive processes*: Academic Press London, 1994.
- [6] S. España, et al., "An empirical comparative evaluation of requirements engineering methods," *Journal of the Brazilian Computer Society*, vol. 16, pp. 3-19, 2010/05/01 2010.
- [7] K. L. Norman and E. Panizzi, "Levels of automation and user participation in usability testing," *Interacting with Computers*, vol. 18, pp. 246-264, 2006.
- [8] A. Cairo, *The Functional Art: An introduction to information graphics and visualization*: New Riders, 2012.
- [9] Coursera. Available: <https://www.coursera.org>
- [10] B. B. Bederson, et al., "Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies," *ACM Trans. Graph.*, vol. 21, pp. 833-854, 2002.
- [11] M. A. Russell, *Dojo: the definitive guide*: O'Reilly, 2009.
- [12] C. Reas and B. Fry, *Processing: a programming handbook for visual designers and artists* vol. 6812: Mit Press, 2007.
- [13] M. Bostock, et al., "D³ Data-Driven Documents." *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, pp. 2301-2309, 2011.
- [14] F. van Ham, "Re-inventing and Re-implementing the Wheel. Visualization Component Reuse in a Large Enterprise," in *SIGRAD*, 2012, p. 5.

[15] K. V. Soares Sousa, Jean; Henderson-Sellers, Brian; Gonzalez-Perez, Cesar, "Evaluating a graphical notation for modelling software development methodologies," *Journal of Visual Languages and Computing*, vol. 23 (4), pp. 195-212, November 2007 2012.

[16] Z. Ning, et al., "OncoViz: a user-centric mining and visualization tool for cancer-related literature," presented at the Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, 2010.

[17] O. Pastor, et al., "OO-Method: An OO software production environment combining conventional and formal methods," in *Advanced Information Systems Engineering*, vol. 1250, A. Olivé and J. Pastor, Eds., ed: Springer Berlin Heidelberg, 1997, pp. 145-158.

[18] P. Molina, et al., "JUST-UI: A User Interface Specification Model," in *Computer-Aided Design of User Interfaces III*, C. Kolski and J. Vanderdonckt, Eds., ed: Springer Netherlands, 2002, pp. 63-74.

[19] I. Jacobson, et al., *The unified software development process* vol. 1: Addison-Wesley Reading, 1999.

[20] N. Nunes and J. Cunha, "Wisdom — A UML Based Architecture for Interactive Systems," in *Interactive Systems Design, Specification, and Verification*, vol. 1946, P. Palanque and F. Paternò, Eds., ed: Springer Berlin Heidelberg, 2001, pp. 191-205.

[21] F. Valverde Giromé, "Oows 2. 0: un método de ingeniería web dirigido por modelos para la producción de aplicaciones web 2. 0," Research Center on Software Production Methods (PROS) Universitat Politècnica de Valencia, Valencia, Spain, 2010.

[22] Q. Limbourg, et al., "USIXML: A language supporting multi-path development of user interfaces," in *Engineering human computer interaction and interactive systems*, ed: Springer, 2005, pp. 200-220.

[23] E. Gamma, et al., *Design patterns: elements of reusable object-oriented software*: Addison-Wesley Longman Publishing Co., Inc., 1995.

[24] M. Brambilla, et al., "A Tool for Model-Driven Design of Rich Internet Applications Based on AJAX," in *Handbook of Research on Web 2.0, 3.0, and X.0: Technologies, Business, and Social Applications*, ed: IGI Global, 2010, pp. 96-118.

[25] P. Martín-Rodilla, "The role of software in cultural heritage issues: Types, user needs and design guidelines based on principles of interaction " in *RCIS'12: Sixth International Conference on Research Challenges in Information Science*, Valencia, Spain, 2012, pp. 1-2.

[26] C. Gonzalez-Pérez and C. Parcerio-Oubiña, "A Conceptual Model for Cultural Heritage Definition and Motivation," in *CAA'11: 39th Annual Conference on Computer Applications and Quantitative Methods in Archaeology* ed Beijing, China, 2011.

[27] M. Doerr, "The cidoc crm, an ontological approach to schema heterogeneity," *Semantic interoperability and integration*, vol. 4391, 2005.

[28] C. Gonzalez-Perez, et al., "Extending an Abstract Reference Model for Transdisciplinary Work in Cultural Heritage," in *Metadata and Semantics Research*, vol. 343, J. Dodero, et al., Eds., ed: Springer Berlin Heidelberg, 2012, pp. 190-201.

[29] C. González-Pérez and C. Hug, "Crafting Archaeological Methodologies: Suggesting Method Engineering for the Humanities and Social Sciences," in *Computer Applications and Quantitative Methods in Archaeology (CAA) 2012*, Southampton, UK, 2013.

[30] P. Martín-Rodilla, "Software-Assisted Knowledge Generation in the Archaeological Domain: A Conceptual Framework," presented at the 25th International Conference on Advanced Information Systems Engineering (CAiSE 2013): Doctoral Consortium, Valencia, Spain, 2013.

[31] F. Criado-Boado, et al., "Before the Barrows: Forms of Monumentality and Forms of Complexity in Iberia and Uruguay," in *Archaeology of Burial Mounds*, University of West Bohemia and D. o. A. L. Š. (ed.), Eds., ed, 2006, pp. 38-52.

[32] "Interaction Flow Modeling Language."

[33] C. Alexander, "The timeless way of building. 1979," ed: New York: Oxford University Press.

[34] Y. D. Network. (2013). Yahoo Design Pattern Library. Available: <http://developer.yahoo.com/ypatterns/>

[35] P. J. Molina Moreno, "Especificación del interfaz de usuario: de los requisitos a la generación automática." Universitat Politècnica de València, 2003.

[36] R. Wieringa, "Design science methodology: principles and practice." presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2, Cape Town, South Africa, 2010.