Interface Reconfiguration Techniques in Interactive Data Mining and Processing

Wolfram Luther¹, Benjamin Weyers¹

Dept. of Computer Science and Cognitive Science, University of Duisburg-Essen, Germany,
¹{luther, weyers}@inf.uni-due.de

Abstract

Data mining tasks are often correlated with complex interaction processes. In the context of data mining software, two perspectives are of particular interest. The first is concerned with the representation of data for visual or aural cues. The second meets the interactive data mining process via an appropriate user interface that offers all necessary operations in a usable and effective way. Data mining processes are often context- and user-dependent processes that differ in the type and complexity of task and the amount of data as well as in the nature of the user’s involvement. This paper describes a new approach to interactive data mining based on Concept Keyboards (CK) and user interface reconfiguration methods.

1 Introduction

Interactive data mining and data processing techniques combine various requirements and characteristics that have to be taken into account when developing an effective and efficient user interface. Users who solve data mining or data processing tasks using specific software are often confronted with many different operations and configuration and visualization options offered by the software or algorithms. As a result, users are interested in defining their interface in a way that meets their individual needs and requirements or even makes an interactive data mining process possible. The interface customization on one hand and the adaptation to specific types of data and information visualization on the other reduce the complexity of the interaction and help avoid errors arising from interaction.

The interaction between the user and the mining tools can be implemented using a traditional keyboard and mouse device. On the conceptual level, the keys would neither give the user any idea of the functionality that will be executed by pressing them nor provide him with the possibility to reconfigure or extend a given initial keyboard. CKs are an example of individual user interface solutions that can also be used for data mining purposes based on the traditional keyboard metaphor [1,2]. CKs are configurable keyboards in which each key can trigger a well-defined step of an algorithm in the data mining process, a navigation option within a data set or other tasks in interactive data mining. Possible implementations in hard- and software have been successfully used several times in recent research and development (cf. References in [1]). Figure 1 shows two different types of CKs for learning complex algorithms and protocols: a hardware implementation on the left side and a software implementation on the right.

2 State of the Art

Especially in teaching and learning systems, CKs have proven their merit as helpful devices enforcing the learning success of students resulting from their high usability and adaption to the student’s cognition. They can be easily generated via a user interface markup language like UIML, XUL etc. Different case studies and evaluations have shown that various user groups,
including people with reduced visual cues, were able to train cognitive skills using learning software together with CKs [2,3,4].

In recent years, several researchers have examined the interaction aspect between user and data. Q. Chen, X. Wu and X. Zhu developed OIDM [5], which provides three categories (classification, association analysis and clustering) of data mining tools and interacts with the user to facilitate his/her mining process. The authors in [6] introduced a user-centered three-layered framework consisting of a philosophy layer, a technique layer and an application layer. The philosophy layer focuses on multiple views of concept identification and formation using various formal languages. At the technique layer, multiple strategies for normal and abnormal situation handling and pattern discovery methods are addressed. The application layer is concerned with explanation-oriented data mining.

In [7], the authors proposed adaptive solutions for metadata navigation and filtering in a complex, heterogeneous environmental database and warehouse. They studied the GUI design for complex environmental data sets. The interfaces served data retrieval and complex information visualization tasks. However, to the best of our knowledge, the role of CKs as input devices in interactive data mining has not yet been addressed.

Fig. 1 A CK for the AVL algorithm and the Wide Mouth Frog protocol. The keys on the hardware keyboard (MoveUp, MoveLeft, MoveRight, and MoveToRoot) allow navigation on the binary tree; the software keyboard contains the operations to execute the computer network authentication protocol.

3 Tasks in Interface Reconfiguration and Redesign for Data Mining Applications

In the context of interactive data mining applications, various tasks come up for interface reconfiguration and redesign. The goal is a full bidirectional interaction process that integrates input and output operations (e.g., data selection, filtering, representation and visualization options) into software for data mining. Scientific work in the context of interaction with data mining application and data in general lacks identifying requirements and solutions on the input side of the user interface. In this contribution, we will identify several issues for interface reconfiguration and redesign for interactive data mining applications.
3.1 Input Devices

We will now present a rough classification of tasks in interactive data mining together with corresponding buttons on the input device and their concepts. The following table gives a review.

<table>
<thead>
<tr>
<th>Task</th>
<th>Physical aspect of key</th>
<th>Concept associated with key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text input/retrieval (3.4)</td>
<td>Button with individual shape and impression indicating the semantics of the triggered action</td>
<td>Access to a template for text input</td>
</tr>
<tr>
<td>Data input support (3.3)</td>
<td>Button triggering a functionality</td>
<td>Access to file directory or data list</td>
</tr>
<tr>
<td>Filtering of data/data selection (3.3)</td>
<td>Labeled button</td>
<td>Selection from data set by using a special selection tool/instrument</td>
</tr>
<tr>
<td>Selection/execution of mining algorithm (often combined with parameter input) (3.4)</td>
<td>Labeled button (Association, classification, clustering, exploration, feature extraction, similarity)</td>
<td>Launching an algorithm</td>
</tr>
<tr>
<td>Parameter tuning (3.5)</td>
<td>Labeled button or slider</td>
<td>Defining or changing parameter values</td>
</tr>
<tr>
<td>Selection of representation techniques concerning data or metadata (3.4)</td>
<td>Labeled button that displays sound or graphics</td>
<td>Configuration of information output, e.g., image list selection, 1D-, 2D-, 3D-, nD-vector data, 2D plots, multiple 3D views, color mode selection. Typical statistical representation of histogram-based data</td>
</tr>
<tr>
<td>Navigation in structured data (3.2)</td>
<td>Arrow button or labeled button</td>
<td>Moving or setting the focus on a node of the hierarchical structure</td>
</tr>
</tbody>
</table>

3.2 Navigation in Structured Data

In 2007 we developed a visualization system for Bayesian networks in a web application which may serve as an example of interactive data mining for graphical data. Navigation in graphical data structures is often a problem when using only the basic keyboard setting. A special operation, such as expanding or navigating from node to node, is often hard to match directly to a classic keyboard. Using a CK is much easier because any graph-specific selection (e.g., select sibling, select all siblings, select parent etc.) and navigation operations (e.g., move focus to selected node, move to parent, move to leaf, expand sub graph etc.) can be directly mapped to the keyboard using meaningful icons and inscriptions.

3.3 Filtering of Data Subsets, Data Selection

Interface reconfiguration and redesign in the context of CKs is a suitable solution for data mining processes in surgical planning and medical image-processing applications. In the European project PROEOP, for instance, we developed a tool for surgical planning [8]. In the context of this application, a selection tool was implemented for the definition of point subsets from a point cloud representing the femur bone (cf. Figure 2). During development, problems in using this tool were identified as caused by offering only the mouse as an input device for positioning the tool. It has been shown that the keyboard is more efficient and precise. Still, the tool has to be positioned in a 3D environment, which makes navigation in third dimension (z-axis) necessary. CKs offer the opportunity to add this functionality to an interface in an intuitive manner. Diagonal translation operations as well as rotation operations can be added to the same keyboard easily, thus extending the functionality of the input interface and advancing the usability of the keyboard by adapting the key positioning to the user.
Fig. 2 Selection tool based on superquadric shapes for selection of subsets of 3D point clouds. The figure shows various configurable shapes of the tool (shown in the upper right corner in the initial configuration) that can be manipulated by a CK.

Other examples in this field of applications are complex systems for processing large amounts of medical image data (PACS). This image data is normally generated by x-ray, CT, MRI etc. and used for surgical planning and diagnostics using a number of different data mining tools in 2D and 3D. These pieces of software offer a large number of different data mining tools and algorithms for manual and automatic selection of blood vessels, organs and so on. Discussions with producers of PACS and doctors have shown that classical keyboard- and mouse-based interaction often does not offer enough opportunities for interaction with these tools because of the limited number of keys and key combinations. In this context, extending basic input devices using CKs allows these tools to be matched to new keys, replacing complex key combinations and extending the number of keys on a classic keyboard. Not least, CKs can also be adapted individually to every doctor using the system.

Modern imaging techniques, like MRI, generate 3D images raising the same problems for filter-

Fig. 3 Interactive tool for training stochastic distance measures for retrieval of spelling invariants in nonstandard texts. On the left side is a multidimensional scaling view, and on the right an interface for training the retrieval algorithms.
ing and selection techniques as discussed above. These problems can also be appropriately addressed by CKs by extending mouse-based interaction with complex navigation, selection and filtering operations on 3D image data.

3.4 Selection of Representation Techniques Concerning Data or Metadata

In a further application, InterVerdiKom [10] (Reliable traffic modeling in multi service networks and analysis of resource requirements for Quality of Service support), the CK gives the user access to the navigation tree representing the whole modeling and analyzing process, whereas the output screen is also used to display the process structure tree, a console for numerical results and further graphical output of special modeling and analyzing software. In addition to direct interaction with data, another task of interactive data mining is manipulation of the representation. Again, the selection depends on what type of data is being presented and the user selecting the representation.

A recent paper entitled Retrieval of Spelling Variants in Nonstandard Texts Automated Support and Visualization [9] presents an interactive tool for training stochastic distance measures. These measures are used to find spelling variants at a certain distance from the standard spelling in old German texts (cf. Figure 3). The user interface is separated into three main views: The Histogram View allows for an overview of thousands of data items, the Multidimensional Scaling View is used to display sets of several dozen to a few hundred and the Table View can display different levels of detail and be used to interactively modify the replacement costs of the distance measure via the CK that is shown on the right side of Figure 3.

![Diagram](image)

**Fig. 4** Bidirectional workflow for creation of Bayesian networks for interpretation of gene expression data.
3.5 Parameter Tuning (Learning Algorithms)

Data mining algorithms often profit from learning capabilities. As described above, searching processes on nonstandard texts have to learn because basic spelling rules are not sufficient. Therefore, algorithm parameters have to be manipulated through an adequate interface. There are two factors that influence the reconfiguration of an input interface: the type of algorithm and the type of data to be learned from. Using a CK would be an adequate solution (cf. Figure 3, right side).

3.6 Conclusion

A bidirectional interface enriched with methods for interface reconfiguration builds an ideal basis for successful interactive data mining applications. The example in Figure 4 shows a workflow from bioinformatics describing the creation of a Bayesian network. The former workflow using pen and paper and the various types of software require many conversion steps and thus allow for many errors. A piece of software called BINViz [11] integrates this workflow into a unique interface. Extending this tool with a CK, as shown in Figure 5 on the right, enriches the integration such that data mining tasks can be accessed using one coherent input device. The CK implements various operations for creating and navigating in Bayesian networks as well as steering conversion and data processing algorithms. The goal for future work is, therefore, to complete the list of tasks, describe patterns for possible solutions using CKs and implement a general tool for the design and redesign of a user interface that meets users’ adaptation and usability requirements.

4 Formalization and Software Realization

A flexible implementation for CK creation and reconfiguration requires a formal basis to describe the physical appearance of the CK, on the one hand, and its behavior—also called interaction logic—on the other. Through this consideration of a user interface, it is necessary to model these two perspectives using visual languages and then translate this semi-formal description to a formal one. Thus, we are developing a framework for the formal description and modeling of user interfaces based on well know formalisms meeting these two different perspectives, as shown in Figure 6: the visual modeling perspective and the formal machine-based perspective. For visual modeling (semi-formal) we are using a BPMN [12]–based visual language to describe process-based interaction logic like that shown in Figure 7. The physical representation of a user interface is modeled using the physical interaction elements (widgets) themselves. For machine interpretable (formal) description of a user interface, this visually modeled user interface is translated into formal languages like UIML [13] or UsiXML [14] for the physical representation as well as co-
Fig. 6 Overview of the framework developed for formal modeling and description of user interfaces.

Fig. 7 Example of interaction logic for a button widget.

5 Conclusion and Outlook

User interfaces for data mining software often concentrate on the data representation aspect, rather than on the input and thus the integration of input and output into one single, bidirectional interaction workflow. We have shown that this aspect of reconfigurable and redesignable input interfaces (CKs) can be a benefit for data mining software. We also have introduced new software for designing and reconfiguring user interfaces for tasks in interactive data mining.

Resulting from the various issues presented in this paper, future work should address the implementation and evaluation of the reconfiguration and redesigning software as well as extend the basic approach to interactive data mining software presented in the second and third sections above.

lored Petri Nets [15] formalizing the interaction logic of a user interface. To complete the framework, we introduced a tool support layer by developing a piece of software for formal design and redesign of user interfaces (the UIEditor) as well as tools for rendering physical representation and for interpretation of formal interaction logic.

The UIEditor implements two different views. The first view implements an editor to visually design and redesign the physical representation of the user interface (cf. Figure 5). The second view (cf. Figure 7) offers a visual editor for reconfiguring the interaction logic that describes the possible interaction with the underlying system. Through this separation of physical and logical representation, the user is able to describe and model the interface in a highly flexible manner.
References