Visual Analytics for Domain Experts: Challenges and Lessons Learned

Clemens Arbesser*
VRVis Research Center

Thomas Mühlbacher†
VRVis Research Center

Stefan Komornyik‡
HAKOM Solutions GmbH

Harald Piringer§
VRVis Research Center

Abstract

In many fields, domain experts can benefit from applying visual analytics solutions to their tasks and problems. Visual approaches are often necessary for finding anomalies or structural changes in the data, providing comprehensive overviews, formulating and testing hypotheses, and many more. However, domain experts are rarely visualization experts themselves. Off-the-shelf visualization software is usually too powerful and complicated or too simple for their professional needs.

In this paper, we present a strategy for creating powerful yet easy-to-use visual analytics solutions for specific problem domains and users. We also discuss challenges and report lessons learned from developing visual analytics dashboards in a variety of domains for almost three years.

Index Terms: I.6.9.f [Simulation, Modeling and Visualization]: Visualization—Visualization Systems and Software

1 Introduction

Domain experts often face challenging problems that are hard to solve automatically. This is an opportunity for visual analytics applications [2].

However, we found that in practice, many experts do not regularly use visualization software. Our observations are based on collaborations with partners in the energy sector, the public healthcare sector, the automotive industry, industrial manufacturing processes and others. Many domain experts write their own scripts (e.g., in Python, Matlab, Excel, ...) and generate visualizations only when absolutely necessary. These visualizations usually suffer from a number of limitations and disadvantages. To name but a few, these include:

- The chosen visualizations may not be suitable, since domain experts are not necessarily visualization experts
- Interactions with visualizations are usually extremely limited, if available at all
- Creating effective visualizations requires considerable amounts of valuable domain expert time

According to feedback from partners and domain experts, this is usually the result of a mismatch between off-the-shelf software and the specific needs of domain experts. Powerful visualization toolkits like Tableau [11] or Spotfire [12] are often targeted towards typical business intelligence data and use-cases, but not towards specialized use-cases (e.g., model validation in energy sector). Furthermore, these general purpose systems do not automatically take into account domain-specific terminology or policies. Perhaps most important of all, domain experts typically do not want to design visual dashboards themselves.

Therefore, we argue for designing domain-specific dashboards built on top of general purpose visualization toolkits. In previous work, we presented 'task-tailored dashboards' [6] and case studies in the energy sector [5] and automotive industry [7]. In this paper, we want to improve and generalize our strategy for deploying domain-specific visual analytics solutions. As such, our contributions comprise:

- A summary of challenges faced when designing dashboards for domain experts
- A generalized strategy to address these challenges, as well as a sample use case
- A summary of lessons we learned over the course of several years

2 Challenges

We have previously argued in favor of 'task-tailored' dashboards instead of comprehensive visualization toolkits. This shifts the development focus to the dashboard design rather than feature implementation. However, designing dashboards for domain experts is hard for several reasons. Most obviously, it requires the visualization experts to have some understanding of the target domain. This is usually accomplished by performing an a-priori task analysis in joint sessions with domain experts. While time-consuming, this usually works reasonably well. However, there are a number of other challenges that are more difficult to solve.

2.1 Make the software easy-to-use, but powerful enough

This is an obvious, but difficult problem, which is exacerbated when designing dashboards for specific tasks rather than comprehensive frameworks. Some of the most common design questions are: How many views are appropriate? How many views should be visible at any one time? Should feature X be enabled in dashboard Y? How can I guide users to the information they are looking for?

2.2 Minimize the overhead for users

Using the visual analytics dashboards should be as easy as pressing a button. Ideally, domain experts should not have to worry about any of the following:

- How can I launch the dashboards?
- How do I import the data?
- How do I configure this view to always show data from X in blue? (and similar policy-related adjustments)
- How can I export my findings to {Excel, ...}?

2.3 Support tasks in the scope of workflows

Specific tasks which are addressed by dashboards are usually in the scope of larger workflows. For example, an analyst may want to optimize a forecast model for a time series. This involves several tasks (dashboards), but it is not clear how these dashboards should be connected to support the user in their workflow. In other words,

*e-mail: arbesser@vrvis.at
†e-mail: tm@vrvis.at
‡email: Stefan.Komornyik@hakom.at
§e-mail: hp@vrvis.at

dashboards should be linked in a way that makes them applicable not just to standalone tasks, but to larger workflows as well.

3 Approach: Visual Analytics for Domain Experts

Using established design study practices [9] and empirical evaluation approaches [3], we have developed a generalized strategy for creating visual analytics solutions for domain experts (Fig. 1).

As a first step, domain experts and dashboard designers jointly analyze existing workflows to identify recurring tasks and existing tools and practices. Workflows usually comprise several tasks. For example, the workflow 'create forecast model' may contain separate tasks for 'rank feature dependencies for target feature', 'investigate feature dependency in detail' and 'create/optimize model for target'. The goal is to split workflows into small but meaningful tasks. In particular, each task should be meaningful in their own right, even outside their originating workflows.

Day-to-day practices must be evaluated for several reasons. First, dashboards should be designed to be natural extensions to existing tools. Second, dashboards should reflect expectations by specific domain experts. For example, '*anomalies are purple*', or '*Ctrl+C exports the current selection to the clipboard*'. Seemingly benign discrepancies can make all the difference for domain experts.

Once tasks and domain specifics are identified, the visual analysis experts can start designing dashboards. Each task should be covered by exactly one dashboard. Dashboards should be designed and tested both as part of a workflow and on their own. In particular, the outputs of earlier dashboards in the workflow should be inputs to later dashboards in the same workflow. This allows smooth transitions between dashboards, without the need for user intervention.

We follow the design activity framework [4] to guide our iterative design process. Early iterations often lead to overly complex designs. A good indicator for this is the number of visualizations offered by a dashboard. In our solutions, we rarely show more than 3 visualizations at any time by default. We do not claim general applicability of this heuristic. The domain expert is always the ultimate arbitrator.

Finally, the dashboards are integrated into existing tools and then deployed. Software integration may be as simple as offering additional context menu entries or buttons in existing tools (e.g., 'Start Visual Analysis' macro button in Excel). However, in many cases, this involves not only domain experts and dashboard designers, but also IT support on both sides. One of the hardest tasks in this regard is the import and export of data. Use of common file-exchange formats can help, but is not always supported by existing tools due to the use of proprietary formats and protocols.

4 Use Case

As a case-in-point, this section illustrates (part of) an example workflow in the energy sector by using 'Visplore' dashboards. 'Visplore' is a comprehensive visual analysis toolkit developed at the VRVis Research Center. It is a multiple linked views system and supports many visualization techniques for multivariate data (scatter plots, histograms, parallel coordinates, ...) as well as some non-standard views (e.g., an adapted version of the rank-by-feature framework [10, 8], a dedicated data quality view [1], etc.). 'Visplore' has been used in a variety of domains since 2004, including the energy sector, the public healthcare sector and industrial production. Recent additions allow the creation of task-tailored dashboards [6].

In this example, the data comprises hourly measurements of several photovoltaic power stations for one year ('PV_01' = photovoltaic power station 01). Meteorological data such as temperature, global radiation, air pressure etc. have been acquired for the same time period. The names of data columns have been anonymized in this data set.

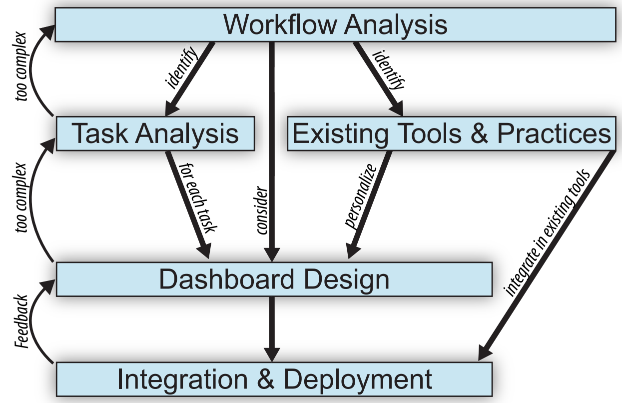

Figure 1: Generalized strategy to develop visual analytics solutions for domain experts. We consider domain-specific personalization and integration in existing tools as one of the most important steps for successful adoption by domain experts.

The data analyst, an expert in the energy sector, is tasked with creating forecast models for each photovoltaic power station. She imports the data into her modeling software and clicks on a button to launch the 'Dependencies' dashboard (Fig. 2a). This dashboard allows the selection of a target feature (in this case: 'PV_03'). All other features are then ranked by relevance for the selected feature (in this case using the Pearson coefficient). In this example, 'Global_Radiation_8' is at the top of the list. The adjacent scatter plot shows the strong linear correlation in detail.

As a next step, the analyst decides to investigate whether this correlation is equally strong throughout the year. She switches to the 'Correlations' dashboard by double-clicking the respective entry on the left side. This dashboard automatically shows the currently selected correlation (Fig. 2b). Drilling down by month (top right), the analyst discovers that the correlation in January is significantly lower compared to other months.

She selects the bad fitting data (Fig. 2c) and switches back to the 'Dependencies' dashboard. Ranking features by their relevance for the selected data reveals that 'Global_Radiation_5' is a better fit for this data subset. To create a forecast model, she decides to use 'Global_Radiation_5' for January and 'Global_Radiation_8' for all other months.

5 Lessons Learned

VRVis is involved in an ongoing process with several partners to develop dashboard solutions for domain experts. One of our longest running partnerships is with HAKOM Solutions, an IT-service provider in the energy sector. HAKOM distributes a platform for time series management and forecasting to over 40 client businesses. 'Visplore' dashboards have been part of this platform since late 2014, but additions such as support for workflows via dashboard switching are much more recent (summer 2017). Most dashboards have by now run through several design iterations. Details on some of the dashboards and their application can be found in previous work [5, 7]. In this section, we summarize the lessons we learned for designing and deploying dashboards for domain experts.

5.1 No size fits all

To visualization experts, many problems of domain experts appear clearly related. For example, domain experts in the energy sector are often faced with 'correlation' problems (see Sec. 4). But similar problems arise in many different domains as well. For example,

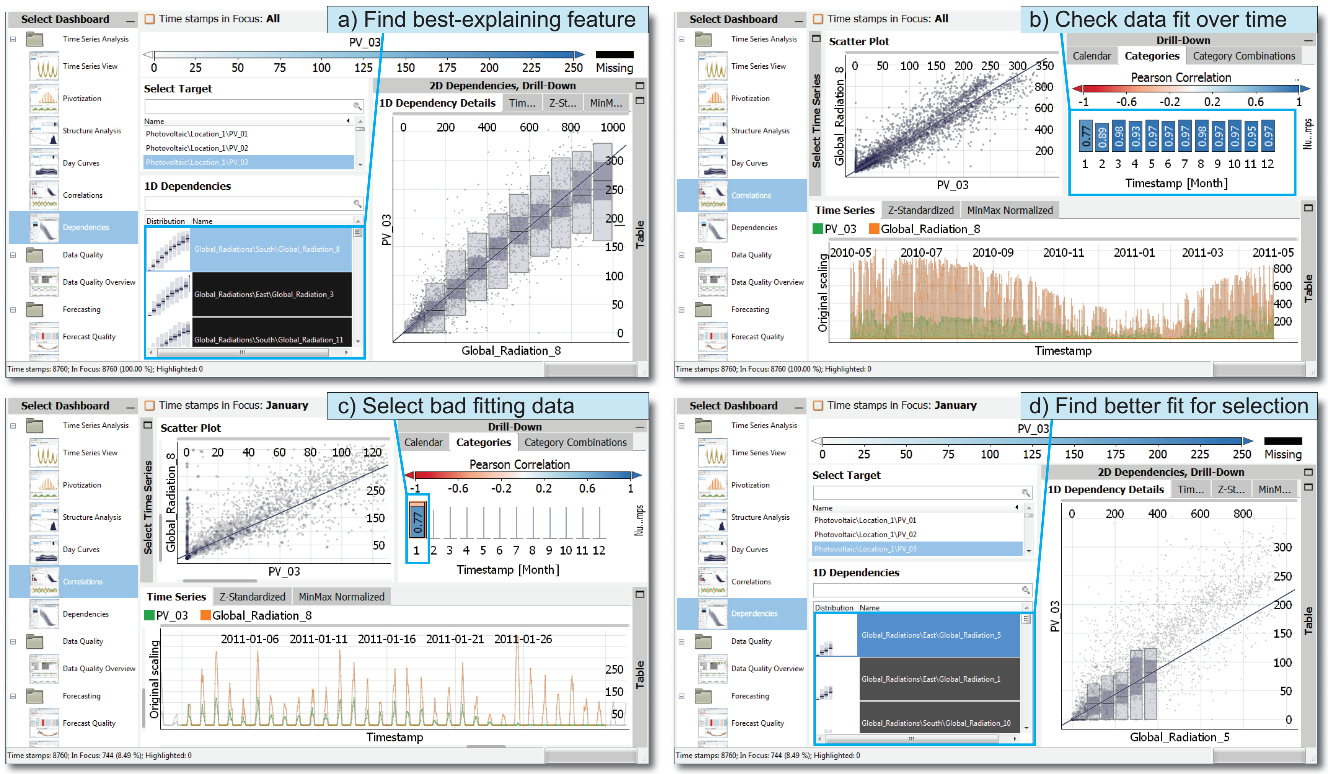

Figure 2: Example dashboard workflow to model photovoltaic power production. The analyst first starts the dashboard 'Dependencies' and selects the best-explaining feature for power plant 'PV_03': Global_Radiation_8 (a). She switches to the 'Correlations' dashboard and discovers that for January, the goodness of fit is significantly lower compared to the rest of the year (b). She selects January (c) and switches back to the 'Dependencies' dashboard. She discovers that for the selected data, 'Global_Radiation_5' is a better fit (d). To optimize model performance, she considers creating a composite model for 'PV_03' involving both 'Global_Radiation_5' (January) and 'Global_Radiation_8' (rest of the year).

an expert in the public healthcare sector might be interested in the correlation between the frequency of certain diseases and public healthcare indicators such as the availability of vaccines.

We have often been tempted to develop a single 'correlation' dashboard for deployment in all domains. However, we have since come to understand how difficult or downright impossible such an endeavor really is. The main difficulty lies in the many differences not just between problem domains, but between business practices and the domain experts themselves. Different data require different visualizations, and different workflows require subtle but sometimes far-reaching design changes. While re-using core parts of dashboards in other contexts is usually possible and desirable, the final solution always has to be adapted to match domain experts' expectations.

5.2 Size matters

When designing dashboards, one of the most difficult design decision involves the number and the arrangement of views. This is ultimately a trade-off between flexibility and simplicity. Extensive and repeated feedback sessions with our partners have led us to believe that in almost all cases, simpler dashboards are better. Advanced features should be hidden by default (e.g., behind a 'show advanced' option). For expert users, a separate 'advanced' version of a dashboard can be deployed in addition to the 'regular' variant.

5.3 No integration, no adoption

Without proper integration of dashboards into target domain expert software, using them becomes difficult and cumbersome. In particular, this involves the following:

- Use of familiar terms and policies in dashboards (e.g., calling data columns 'time series' or using a specific color for 'faulty' data)
- One-click launch of dashboards from within target software
- Automatic import of relevant data
- Automatic parametrization of dashboards according to context (e.g., step in workflow) to provide guidance
- Automatic storing and restoring of dashboard-specific settings to support preferences for different users and workflows
- Automatic formatting of exports according to existing policies

Failure of adoption by domain experts is just as likely due to issues of integration rather than issues of software complexity. In practice, we found poor integration to be one of the main reasons why many domain experts rarely use visualization toolkits.

5.4 Scripting > Programming

Designing dashboards in low-level programming languages like C++ would be prohibitively slow. As such, external APIs for visualization frameworks are a necessity for efficient dashboard design. They may also be required for integration purposes. In practice, we only program core functionality in C++, but write the entire dashboard logic in Python. All of our dashboards are completely specified using a combination of XML and JSON files as well as Python scripts for dashboard-specific features and GUI controls. External APIs also allow automating the data import, which we hide from the user by default.

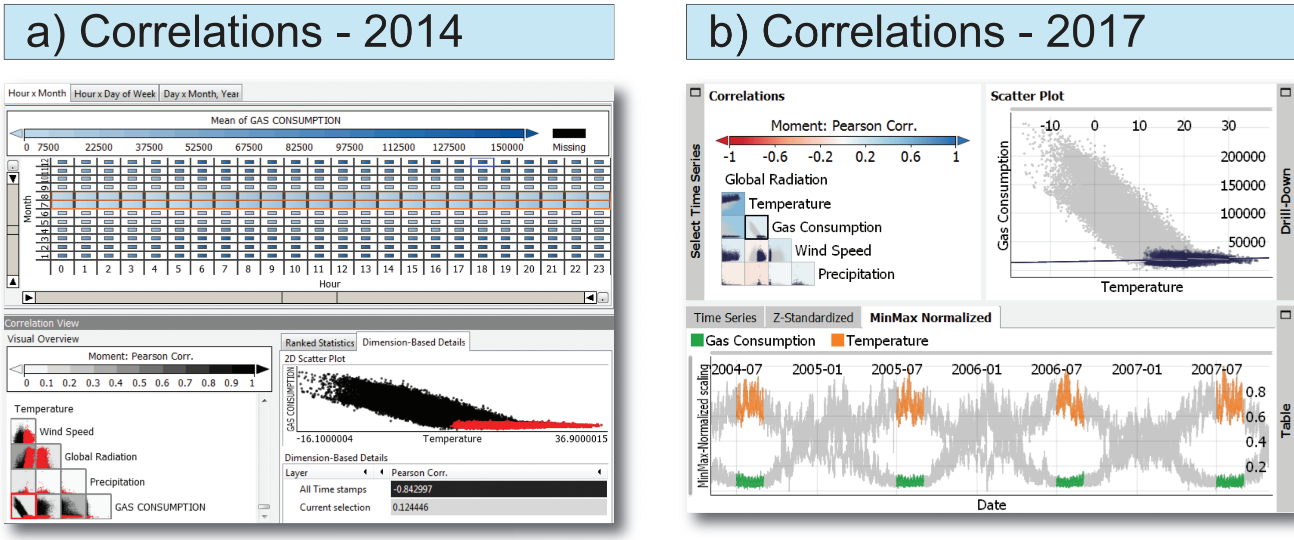

Figure 3: Dashboard design is an iterative process. This example shows how the dashboard 'Correlations' changed since its first design iteration in 2014 [6]. Apart from obvious changes to the layout of the dashboard, there are a host of more subtle changes as well. For example, most views can now be expanded or minimized by clicking the respective icons in their title. Many control elements such as the data mapping controls surrounding the calender in (a) are now hidden. Correlations are not just visualized in a scatter plot, but their constituents are also shown as time series in the bottom view of (b).

5.5 Practice makes perfect

Dashboard design is an iterative process. Effective dashboards are always created in collaboration with domain experts and require time and effort on both sides. No dashboard design of ours has stood the test of practice and time without many design iterations. Figure 3 shows how one of our dashboards, 'Correlation', has changed over the course of almost 3 years. The changes reflect the lessons we learned. Advanced features (such as the the breakdown of values per time categories in Fig. 3a) are now hidden by default. Most GUI elements are only shown on demand when hovering the mouse over the respective views. For domain experts in the energy sector, additional time series views on the bottom of Fig. 3b were introduced.

6 Conclusion

We consider specialized visual analytics software for domain experts an emerging field with many practical applications. Task-tailored dashboards have been shown to be more easily accepted and more frequently used by domain experts compared to comprehensive visualization frameworks [5, 7], a finding which we have since often replicated.

In the future, apart from improving our existing dashboards and designing new dashboards for our partners, there are several limitations that we plan to address. For example, our current dashboards always require human guidance and are not suitable for tasks such as automatic reporting. We also plan to improve workflow support with linked dashboards. In particular, by taking into account provenance information such as a users' most recent actions, it may be possible to infer user intent and parametrize views accordingly.

Acknowledgements

VRVis is funded by BMVIT, BMWFW, Styria, SFG and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (854174) which is managed by FFG. We wish to thank our project partner HAKOM, as well as Stephan Pajer for valuable discussions.

References

[1] C. Arbesser, F. Spechtenhauser, T. Mühlbacher, and H. Piringer. Visplause: Visual Data Quality Assessment of Many Time Series Using Plausibility Checks. *IEEE Trans. on Vis. and Comp. Graphics*, 23(1):641–650, 2017.

[2] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. *Visual Analytics: Scope and Challenges*. Springer, 2008.

[3] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical Studies in Information Visualization: Seven Scenarios. *IEEE transactions on visualization and computer graphics*, 18(9):1520–1536, 2012.

[4] S. McKenna, D. Mazur, J. Agutter, and M. Meyer. Design Activity Framework for Visualization Design. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2191–2200, 2014.

[5] T. Mühlbacher, C. Arbesser, and H. Piringer. Statistical Forecasting in the Energy Sector: Task Analysis and Lessons Learned from Deploying a Dashboard Solution. In *Short Paper in Proc. of IEEE Vis 2015, Visualization in Practice Track*, 2015.

[6] T. Mühlbacher and H. Piringer. Task-tailored Dashboards: Lessons Learned from Deploying a Visual Analytics System. In *Proc. of IEEE Vis 2014, Practitioner Experiences Track*, 2014.

[7] S. Pajer and H. Piringer. Challenges in Creating Task-tailored Analytical Dashboards for the Automotive Industry. In *Poster Proc. of IEEE Vis 2015, Visualization in Practice Track*, 2015.

[8] H. Piringer, W. Berger, and H. Hauser. Quantifying and Comparing Features in High-Dimensional Datasets. *Proc. of the 12th Intern. Conf. on Inf. Vis. (IV 2008)*, pages 240–245, 2008.

[9] M. Sedlmair, M. Meyer, and T. Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Trans. on Vis. and Comp. Graphics*, 18(12):2431–2440, 2012.

[10] J. Seo and B. Shneiderman. A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data. *Information Visualization*, 4(2):pages 96–113, 2005.

[11] Tableau Software. Tableau. `http://www.tableau.com`.

[12] TIBCO Spotfire. Spotfire. `http://spotfire.tibco.com`.