Dynamic Ontology Supported User Interface for Personalized Decision Support

Harald Bosch, Dennis Thom, Geoffrey-Alexeij Heinze, Stefan Wokusch, Thomas Ertl
Institute for Visualization and Interactive Systems
University of Stuttgart
Germany
{bosch, thom, ertl}@vis.uni-stuttgart.de, geoffrey.heinze@gmail.com, stefan@wokusch.de

***Abstract*—European citizens are increasingly aware of the influence of air quality and weather on their health and quality of life. At the same time, more environmental information is freely available through a plethora of websites, dedicated portals, and web services. In order to exploit these data for personal decisions one has to identify, retrieve, and combine the information that is relevant to one's personal situation, planned activity, and information need. Often, this task is hindered by different data formats, display styles and data resolutions. The PESCaDO system is a web-based decision support system addressing this issue. The inquiry to the system, as well as the system's result, can cover a broad range of environmental aspects and personal situations and is therefore quite complex. In this work we present a novel approach on how the system can actively assist users in all steps of the decision making process, especially by enhancing the user interaction. This approach combines an intelligent dialog steering method based on analyzing the domain ontology with flexible, dynamic data visualizations for a situation depending orchestration of data sources. Both aspects have been evaluated in on-line user studies, as well as with an expert evaluation of the whole system.**

***Index Terms*—interactive systems; user interfaces; semantic web; decision support systems; environmental factors;**

I. Introduction

With the broad availability of service-oriented web sites that provide environmental data like weather or air quality information, more and more citizens are increasingly aware of the influence that such data can have on personal decisions regarding their health and quality of life. Access to such information is nowadays provided either statically, or at best through user-defined search, which both might be biased towards what the data provider considers relevant to the user's information need. Making sense of the information for environmental decision support, however, requires more than just retrieving information; instead it needs to be related to other information, taking different perspectives on it, and semi-automatic refinement, in short - it requires analysis.

However, traditional as well as visual analysis [1] systems often tend towards being either powerful, feature-rich software tools for tackling domain-specific tasks or generic construction kits for building personal solutions from abstract operators. In both cases they are tailored to expert users willing to learn the usage of rather complex user interfaces with multiple, potentially coordinated visual perspectives and elaborate interaction mechanisms.

Bringing these two observations together, there is a need for an interactive system suitable for casual users that enables them to exploit available data for supporting their personal decisions. The EU funded project PESCaDO tackles this challenge for the environmental data domain by reducing the complexity through personalized interaction techniques and ontology based user assistance. The resulting web-based system provides end user decision support based on data that is automatically extracted from the web and orchestrated using interactive visualizations. At the beginning of each request, the user's information need can be formulated as an abstract query independently from the available data sources. Here, the user is assisted by an intelligent dialog steering mechanism incorporating user profiles, domain knowledge, and context information to highlight missing input parameters and guide the user towards a serviceable and personalized request.

The user's input is formalized using PESCaDO's Problem Description Language (PDL) and associated with available data sources and the system's codified knowledge in the form of an ontology. Using semantic inference algorithms, relevant data is extracted and interpreted from the web and the most important information is isolated and orchestrated to fulfill the request. Finally, to present the results to the user, the system uses an adaptive mechanism to select an adequate ensemble of interactive and configurable visualizations that try to combine requirements resulting from the request as well as personal user preferences and data driven necessities.

This paper focuses on two aspects of PESCaDO that allow for personalized query and result presentation. First, the overall approach is described in Section III together with a short introduction of the system's aspects that are not detailed further in this paper. Section IV then describes how an intelligent query steering can be realized by accounting for previously submitted information, user profiles, and domain knowledge. Section V will give details on the personalized selection and configuration of visualizations and how the user can interact with and adapt visualizations to fit his particular information needs. The described query and visualization components have been individually evaluated through web-based user studies and the complete PESCaDO system was thoroughly tested by environmental specialists. The results of these evaluations will be presented in Section VI. Finally, Section VII concludes and gives an outlook on future work.

 ISBN: 978-1-61208-232-5

II. Related Work

This section summarizes existing work related to our two main contributions; intelligent user interfaces and dynamic visualization of environmental information.

A. Intelligent User Interfaces

In the context of this work, we define *intelligent user interfaces* as systems that react individually on user input based on background information, e.g., the user's profile, previous input, or domain knowledge. This definition is in line with *intelligent support systems* discussed in the work of Delisle and Moulin [2] but does not necessarily have to be based on machine learning algorithms. They can be grouped into three, not necessarily disjunct, categories inspired by Dryer [3].

1) Guides: support users by providing additional information for their tasks. This guidance may range from formatting information for input fields to user manuals for choosing the correct form. If guides dynamically take into account the available information, they can also be considered 'intelligent'. For instance, the *COACH* system [4] builds an adaptive user model and provides contextual help by commenting on the user's actions. Guides are useful for providing local and not too complex information about the currently focused aspect of the user interaction.

2) Wizards: support users by structuring complex input forms into separate, sequential, and thematically coherent pages. Each page can take the previously provided information into account, e.g., to include or exclude branches of the predefined course of the dialog. In WOLD [5], e.g., a wizard for generating new user interfaces can suggest parameter values. Wizards profit from the fact that only few interactive elements are available at any point in time. However, with increasing task complexity it becomes harder to define thematically coherent but independent subtasks to be grouped in a linear structure.

3) Reactive Systems: can actively influence the current dialog. This is often implemented through the use of *agents* [6] and covers a wide area of applications from saving previous input for automatic input completion, to learning the user's behavior to better adapt dialogs to their needs. An example can be found in the system of Lee et al. [7], which supports tourists in planning a path through cities to meet different sightseeing interests. In *mixed-initiative* systems [8], e.g., as in the work of Frank et al. [9] for trip planning, the active part changes between user and machine. While this can also be achieved by using agents, the clearer role definition leads to less user astonishment.

The intelligent user support for personalized query formulation of PESCaDO is a combination of these principles in the form of a guided wizard that reacts intelligently on the user input using an automatically derived rule set.

B. Dynamic Environmental Data Visualization

Traditional visualizations for environmental data can be seen in many forms and application domains. From popular media, most people are familiar with certain iconic representations of meteorological data, like sun-/cloud-symbols for overall weather conditions. Research in visualization has also introduced additional unconventional and advanced representation forms ranging from large-scale weather statistics to representations of complex simulation models [10]. However, there are just a few visualization approaches that allow representations to be automatically configured and adapted based on data and/or user aspects or that account for inherent uncertainties in environmental data.

1) Adaptive Representations: The foundations for the concept of adaptive, data driven visualization models were laid by Mackinlay [11] who presented the idea of automatic visualization systems (AVS) that generate visualizations intelligently based on relational data structures. He interpreted graphical representations as parts of a graphical language. The APT (A Presentation Tool), presented in his work, applies artificial intelligence methods to choose from representations like bar charts and scatter plots. Recently, this idea was extended [12] to allow for automatic representations for visual analysis within the Polaris system [13] . The authors present the Show Me user experience that allows the user to select alternatives and configure visual variables in small multiples of text tables, aligned bars, stacked bars, line charts, scatter plots, and Gantt chars. The idea to use ontology mappings to decide which visualizations should be chosen to represent given web data was presented by Gilson et al. [14], where domain ontologies and visual representation ontologies were linked through a semantic bridging ontology specifying the appropriateness of a given mapping from data to representation. Although these approaches cover many aspects of common data, they fall short of handling combinations of multiple independent data layers and deciding on appropriate combinations of visualizations to represent them, as needed for the personalized decision support scenario of the PESCaDO system. Furthermore, the integration of user models within the decision path and the possibility to interact with visual configuration parameters is underrepresented in these works.

2) Uncertainty: Some researchers have addressed the problem of visualizing uncertain information. Olston et al. [15] demonstrate in bar charts, scatterplots, and line charts, the need to indicate clearly the difference between statistical uncertainties and bounded uncertainties. Statistical uncertainty describes values that can be distributed over an infinite range with a peak on some expected value. Contrastingly, bounded uncertainty guarantees that values lie within a known interval. They propose to use error bars to indicate statistical uncertainties and a technique resembling an ink smearing effect to show bounded uncertainties. Hengl and Toomanian [16] examine the usage of uncertainty visualizations for map data and demonstrate techniques using whiteness in color and pixel mixtures to indicate the proportion of errors in heatmap representations. These techniques are similar to the ones that were applied in our work. However, they have not been used in association with data and user adaptive presentation techniques, which poses a completely new design challenge.

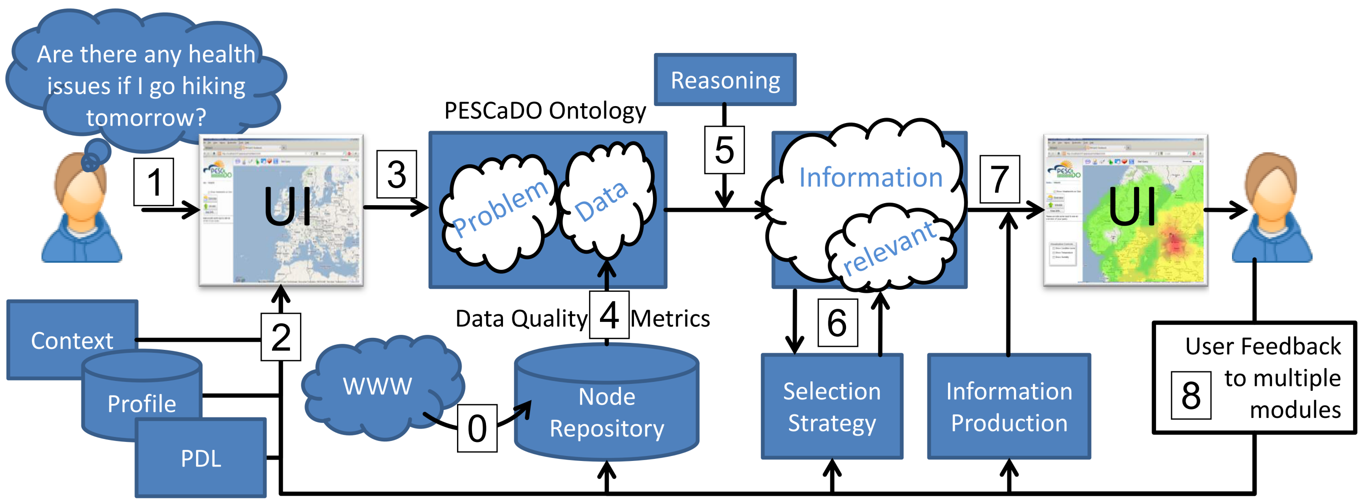

Fig. 1. The decision support loop of the PESCaDO system going from data extraction (0) and the generation of a request (1+2), over the result computation (step 3-6), to a dynamic result representation (7) and interactive feedback (8).

III. Approach

To achieve a combination of simple and fluent user experience together with powerful analysis capabilities, the PESCaDO system builds on two cornerstones: A personalized intelligent query support and a configurable, user tailored result presentation. These components will be described in further detail in the following two sections. Building on these components, the PESCaDO decision support process can be separated into eight steps (see Figure 1 for all steps):

As part of a preprocessing step, a node discovery service identifies web data sources for different time spans, environmental aspects, and geographic regions through keyword spice targeted searching [17]. The resulting resources are categorized, their content is extracted, and the gained information is stored in a repository on a daily basis (step 0). Based on the user's information need (step 1), personal profile, domain ontology, and current session context, an intelligently steered dialog supports the user in supplying the necessary information to complete a serviceable and useful query (step 2). This query is translated into semantic structures using the PDL, which is defined as part of the PESCaDO ontology, in order to validate the user input and associate it with the ontology content (step 3). The semantic structures are then aligned with the available environmental information (step 4) and possible implications are calculated by a general-purpose Semantic Web reasoner (steps 5). Finally, the result is filtered according to a selection strategy (step 6) and presented textually as well as visually in the form of an orchestrated and configurable ensemble of visualizations tailored to user and data needs (step 7). Additionally, the results can be used as the basis for adjustments and feedback in order to refine the query (step 8), if the user's demands should not be satisfied.

Steps 3 to 6 are not described further since it would go beyond the scope of this paper. Instead, we refer to [18], [19] and [20] for more details on the overall PESCaDO approach.

IV. Intelligent Support for Query Formulation

In order to deliver a user interface (UI) that is as generic as the expressiveness of the PDL, but at the same time coherent enough to be comprehensible, the visual interface is personalized intelligently according to available context information and the user's information need. Usually the available context is the set of valid queries, the user profile, the already supplied information, and the execution environment, i.e., the browser of the user. Only as much information as needed to form a valid query is requested to keep the user's effort low.

This work focuses on how to exploit a domain ontology for supporting the user to form meaningful queries. The PESCaDO ontology [21] is compiled from manually crafted scenarios, semi-automatically constructed domain knowledge, and existing domain ontologies. It describes, amongst the relations between activities, diseases, and environmental data, also the valid queries that it can handle. These valid queries are implicitly defined, e.g., through subclass relations, class restrictions, and generic properties (e.g. hasStartDate). Our approach analyzes the ontology for these constructs and automatically generates a simple set of rules that the UI can than interpret during a user session. We thereby use preexisting information to better support the user without manually designing the relationships between the input elements.

On the front end, the UI is designed as an intelligent wizard dialog hovering over a geographical map. This geographic area of interest is the natural link between the query generation and the visual result presentation on the same canvas. The wizard allows a free navigation between the pages in order to avoid patronizing the users. After each user input, the UI evaluates the rule set derived from the ontology to identify forbidden inputs or required fields based on the already stated information. The resulting information is used to highlight the relations between input fields in case of an inconsistent query to enable the users to resolve these issues efficiently.

 ISBN: 978-1-61208-232-5

After the minimum input parameters for the request type are entered, the user can submit her query to the system for decision support. Prior to sending the collected information to the server, the user can review it by inspecting the input parameters on the map (in case of a route or area selection) and a textual summarization of selected data values.

A. Rule Generation

Because every user input needs to be instantiated in the ontology for the result computation, the system has a direct mapping between user input elements and ontology concepts. For each pair of these concepts their ontological relation is examined to infer if they require or exclude each other. Table I lists the relevant class relations that are used to generate the rules. Additionally, *data-type properties* that map to literals instead of concepts, such as *hasStartDateTime*, are examined to create further rules for date ranges and route definitions.

Let us take the first rule of Table I as an example. The subclass of an ontological class is a more detailed definition of the parent class. *Hiking* is a subclass of *Outdoor Activity*. If users state to undertake an *Outdoor Activity*, they have to state one and only one of the subclasses, too. Therefore, a rule is created for each subclass such that: If the parent class is selected and none of its siblings is selected, the subclass is a required input. Because this rule exists for each subclass, they are all required unless one is selected.

Of the roughly 640 concepts in the PESCaDO ontology, 26 were relevant for the UI because they have a mapping to input elements. From their relations and properties, 56 rules were generated and stored in an XML file. The rule's format allows for an easy evaluation in the UI component by substituting the class names for a Boolean, stating if the related variables are filled in by the user. The rule effects are implemented as factors that are multiplied on a fixed standard weight for each input element. Higher weights increase the importance of an input field. Here, *forbidden* has a factor of 0 and *required* has an arbitrary large factor of 1024. Based on additional context data, other rule effects and factors between these two extremes are conceivable to indicate recommended fields.

B. Highlight Guided Input Form

Based on the now known explicit rules, the input fields can be grouped on pages ordered according to the fields' influence. When the most important fields are filled in at the beginning of the interaction, most of the other input fields

TABLE I
THE INFERRED RULES FROM ONTOLOGICAL RELATIONS. HERE, Z_i ARE THE SIBLINGS OF Y AND L IS THE LEAST COMMON ANCESTOR OF ALL Z_i. THE PREFIX *count* IS USED TO COUNT THE OCCURRENCES OF Y.

Ontology Relation	Resulting Rule	Rule Effect
Y subClassOf X	$X \wedge \neg(Z_1 \vee ... \vee Z_n)$	requires Y
S hasSomeValuesFrom $\{Z_1 ... Z_n\}$	S	requires L
not(S hasSomeValuesFrom $\{Z_1 ... Z_n\}$)	S	forbids Y
X hasOnlyValuesFrom(Y)	X	requires Y
X hasExactCardinality(1, Y)	X	requires Y
X hasMinCardinality(2, Y)	X	requires count:Y

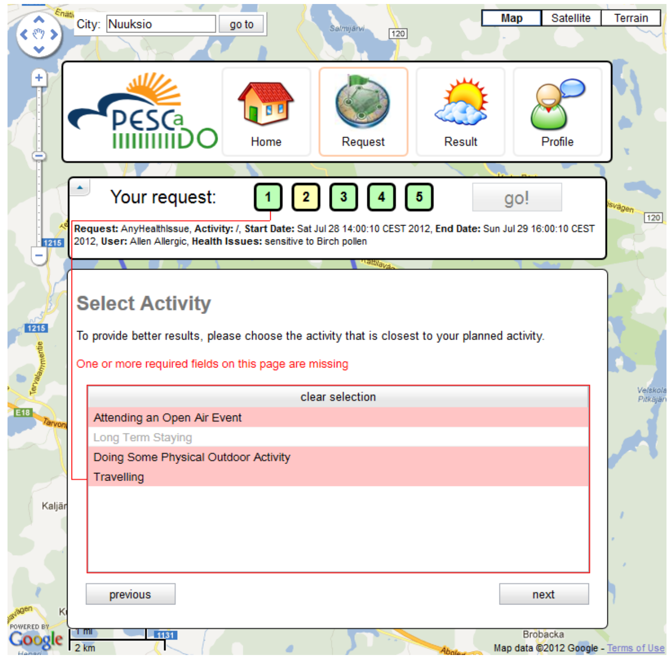

Fig. 2. The PESCaDO query generation wizard. The top row of buttons access different parts of the UI. On the *request* page, four of the wizard's pages have been filled in successfully (green numbers). The current page (yellow) has an error because an input on the first page (red line) requires the user to fill in an activity. A summary of the user-supplied information is given immediately below the page numbers. In the background, a Google Map environment is available for region selection and result visualizations.

should already be marked as required/forbidden by the time the users have to fill them in. In this process, even pages having only forbidden fields are not skipped automatically in order to show inexperienced users potentially available fields and give them the opportunity to go back and change the input that forbid the desired ones. The overall progress within the wizard can be seen by a numbered list of pages at the top of the dialog, which also allows navigating between the pages. Here, the current page and pages with errors are marked by different colors.

Every input field's widget implements its own highlighting methods. This way, even complex and non-standard input methods like a route selection can be highlighted in an optimal way. In the PESCaDO query formulation (see Figure 2), the highlighting of forbidden fields or unfilled required fields follows the common convention to change the color of the fields' borders and backgrounds to gray or red respectively, and including an error message in the vicinity. Additionally, a red line connects conflicting inputs in order to facilitate the tracing of errors. If one of the conflicting widgets is situated on a different page of the wizard, the red line connects to the appropriate page number at the top of the dialog.

The rule effects of the user input are recalculated whenever the user changes a formerly conflicting widget or if the mouse cursor hovers over a navigation control to go to another page of the wizard. If the users did not yet confirm their input, the UI first shows only a warning for the inconsistent input by

 ISBN: 978-1-61208-232-5

marking them yellow. Navigating to different pages confirms the input and then shows the inconsistent input as an error.

V. Result Representation and Feedback

The response of the system is composed of the environmental data that was deemed relevant for the problem and a natural language description of the situation including potential guidelines that refer to the planned activity and the user's profile. To enable the user to interpret the data quickly, a web based environmental data visualization component was developed that employs different visualization techniques to show the environmental situation, embedded in the same interface in which the query was generated. This approach exploits the geospatial nature of the data and also presents the data in the surrounding of the region of interest.

The visualization component is parameterizable to allow for personalization, allow the stacking of visualizations for the concurrent display of different environmental information, and to make concrete values easily distinguishable. It is based on a generic, uncertainty-aware, environmental data model and a set of flexible visualization methods.

If multiple environmental data types are deemed relevant simultaneously, the system decides on the mode (textual vs graphical) in which the environmental information will be displayed. Additionally, it tries to map data types to visual attributes (shape, size, color, position) without delimiting the overall interpretability. In these decisions, the data types, users' profiles, and their previous interactions are considered by a reinforcement learning approach. When users reassign the selected visualization techniques to different data types interactively, this information is send back to the server to further train the mode selection.

A. Geographical Data Model

In order to allow dynamic user adjustments of the parameters and mappings, every visualization technique has to work on the same generic data model. Of course, not every technique is suitable for visualizing a certain data type - e.g. wind direction vectors cannot be visualized with a bar chart display - but the implemented software interface to the data is the same for all data types. It comprises 1) a *data source* component to access the data in a standardized way, and 2) *data objects* that contain the actual data used by the visualization techniques.

1) Data Sources: Environmental information can be provided by web resources and the data node repository in various types, resolutions, dimensions, and with different degrees of uncertainty. Therefore, data sources are used as a mediator that translates and unifies the data into a generic fixed resolution data format. As an example, a temperature heatmap overlay needs to compute a color value for every pixel of the map, whereas the available data providers might only supply one value for each city in the area. The data source offers the functionality to extend these values to the map's resolution by either returning the same value for every pixel within the boundary of a city, or by interpolation.

Each data source provides meta-data about the environmental data that it delivers in order to allow the visualization to adapt itself to the specific data type. Therefore, each environmental aspect has its own set of meta-data like the name of the aspect, its unit of measurement, the specific subtype of data that it returns, and the thresholds for categorizing continuous data. These meta-information can be overwritten on a per user level so that, e.g., the color-coding could be personalized to account for specific sensitivities of a user.

2) Data Objects: Every request for data is answered by a data source with an object of an abstract *data* class. The two main data types are *atomic data* and *complex data*. Atomic data is further divided into having only a single value (e.g. ozone concentration) or an interval of possible values (e.g. min/max temperature). Complex data can be arbitrarily composed (e.g. wind data as combination of single valued wind strength and a wind direction interval) or be a set of similar data types (e.g. air quality as a set of single valued pollutant concentrations). Due to the importance of uncertainty in PESCaDO, each of the data object used in the visualization has an uncertainty score between 0 and 1.

B. Visualization Modes

The visualization component of the system features a set of web-based visualization techniques that are capable of representing environmental data on a map, after they were requested from corresponding data sources. Some of the techniques are tailored to depict a specific data type, e.g. particle flow for wind data, but most are suitable for visualizing multiple types. They can be divided in two main groups; 1) area based visualizations showing continuous data and 2) glyph-based visualizations showing data at sample points using special icons. An example of some of the available visualization techniques can be seen in Figure 3.

1) Area based visualization: Because the spatial extent of the visualization covers the whole area of interest, it can only depict data at one given point in time. If different time steps are relevant (e.g. throughout one day or multiple days) an animation can be used to give an overview and the users can browse through the time steps manually using a slider control.

Well-known examples of area based visualization techniques are *heatmaps* and *isolines*. While heatmaps use color to depict continuous data in two-dimensional areas, isolines depict points on the map where the visualized data matches a predefined value. In our case, these visualizations are drawn semi-transparently over the map and the conversion from data values to color values is based on the thresholds defined in the data source's meta-information. It can therefore be personalized to the current season and each user individually. Isolines are a well-known metaphor for representing atmospheric pressure and use only limited screen space and no color. They occlude minimal information from the background or other data visualizations and can be used in conjunction with color-based views. Both techniques are suitable for showing a broad overview and are mainly applicable to densely sampled data; otherwise the interpolation between the actual data points

Fig. 3. Examples of available result visualizations: a particle flow view of wind data on the left side, a combination of a single point label with a normalized line graph along a route on the middle left, a field of wind data glyphs on the middle right, and a bar chart repeated on multiple points in time along a user selected route on the right side.

can lead to false impressions. This can be counteracted by increasing the translucency if the uncertainty is high.

A special visualization technique for wind data is *particle flow*. Similar to isolines, it occludes little screen space and does not employ color. It is generated by seeding particles at random locations and let the wind data (strength and direction) virtually transport the particles. This transport creates a path that can be drawn on the map and can even be animated for a stronger sensation of the wind velocity.

2) Glyph-based visualization: Glyphs only denote data at singular locations. However, they can be spatially extended by repeating them along a route or in a regular grid within an area. Additionally, they can be moved to a separate view outside of the map while showing data from, e.g., the current location of the mouse cursor on the map.

In the proposed framework, each glyph depicts exactly one data object, but multiple glyphs can be combined to form glyph-based visualizations. This has the advantage that each glyph in the view can depict a different point in time, thereby eliminating the need for exploring a time range manually. For this, an intuitive mapping from geospatial positions to time is needed, which can be found as user-planned routes with timestamps for each waypoint. This simulates the actual travel/hike and shows the data of time when the users will probably arrive at a given location.

Labels and weather icons are common means to convey information to the users. The data value with its unit of measurement, or the appropriate icon, is placed at the specified geographical location, usually - but not necessarily - of a fixed size. An example within the PESCaDO prototype are wind arrows, which map the average wind direction to the rotation of the arrow, the possible directional interval is mapped to an arc in front the arrow, the magnitude is mapped to either size or an animation speed, and uncertainty is mapped to the intensity of a Gaussian blur over the whole icon.

For atomic or composed scalar data - which is the majority of environmental data - (stacked) bars can be used to denote data values to the height of a bar. They can also show negative values by extending below the baseline, as well as intervals and uncertainty by adding error margins or color variations. If bars are repeated along a route, their orientation is changed according to the principle direction of the baseline to avoid overlapping (see last panel in Figure 3).

For displaying multiple data types along a user-defined route, the route itself can function as a baseline for a line chart, which maps the individual data values on sample points of the route as the current height of a line orthogonal to the dynamic baseline. In order to unify the scale of the chart for different data types and simplify the interpretation of data, each value is normalized according to three value ranges for *good*, *medium*, and *bad* values. This qualitative normalized scale is then shown as colored bands behind the lines. The data source's meta-information is again used for the normalization, which allows for user level personalization of the chart.

3) Visualization Manager: All visualization types and available data types are registered at the Visualization Manager component as map overlays and/or separate displays. Some of them offer additional control widgets and legends for integration into the UI. Among the default control widgets is a time slider, which controls the visualized point in time. Also, the visualization manager has its own user interface that controls the mapping from data type to visualization technique for personalization.

VI. Evaluation

The ontology supported query generation framework was evaluated in a web based user study. Two different tasks (plan a hike and get air quality information) to be performed with two alternative versions of the user interface were presented to the users. The task/UI combination and order of the UIs where randomized to avoid learning effects. Besides the presented approach, a previous version of the user interface was used for comparison. This version did not highlight errors dynamically but employed hard-coded error checks prior to sending the user query to the system and hid specific input fields based on defined inputs. During the interaction the time and correctness of results were measured and questionnaires on the user satisfaction and UI preference had to be filled. Overall, 56 participants completed the evaluation. The results showed that the improved UI could be used more effectively and reduces the amount of errors in the submitted queries. Consistently, most users clearly preferred the improved UI. These results were deemed significant by a t-test ($\alpha = 0.05$). A significant speed-up in using the new UI could only be observed during the second task, which means that familiarizing with the scenario is probably the major time constraint.

We also performed a web based user study on the usage of adaptive environmental visualizations amongst 55 participants. Focusing on the aspects of data uncertainty and automatic selection of combined visualizations the participants were asked to solve several information gathering tasks and to rate the systems performance in a questionnaire. During these tests

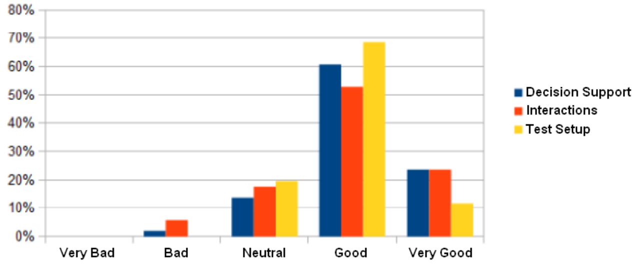

Fig. 4. Results of the adaptive visualizations evaluation. The bars show the users rating of the overall decision support (blue), the interaction methods (red) and the test system (yellow).

we also measured the correctness of the results. In a first phase the users were presented with the individual visualization types in order to learn how to interpret them. Here, two thirds of the participants correctly identified the temperature (3°C tolerance) using only the heatmap. In the subsequent test, the users were shown temperature, wind and air quality data in different combinations (e.g. temperature as heatmap, wind as particle flow, and air quality as bar charts) and they were asked to plan a bicycle ride within a certain time frame and map area. From the questionnaire answers and the correctness of the results we found that the simultaneous display of these data types could be mastered in a coherent fashion that helps the users to get a holistic picture of the situation. The users found the presented methods overall useful and could use them well to solve the presented tasks. The user rating of the information gathering process is shown in Figure 4.

Finally, the PESCaDO system was evaluated by an environmental expert user panel consisting of seven participants. The majority of users deemed the provided information comprehensible and useful and they stated that they would use this kind of service. They found the interface suitable for decision support and pointed out that showing the actual data is important to them. The *particle flow* was not considered to be a good visualization for wind data by some participants.

VII. Conclusion and Future Work

This work presented a novel approach to help end users in formulating complex decision support queries and interpreting the exhaustive results by exploiting domain ontologies and personalization. It is based on externalizing implicit rules that define valid queries in order to guide the user by an improved UI with a coherent structure and dynamic highlighting. The system's results are given in textual and visual mode and provide a personalized overview to suite the visual vocabulary of the user. It therefore allows lay users to employ rich, semantic web based decision support to their day-to-day problems. The visualization component of the approach is tailored to the environmental domain. However, the intelligent support for query formulation could be applied to UIs of other ontology-based systems. In future work, we will examine further possibilities of ad-hoc rule generation based on machine-learning and a unification of error highlighting and default value prediction.

Acknowledgment

This work was funded by the European Commission as part of the FP7-project PESCaDO (FP7-248594).

References

[1] J. Thomas and J. Kielman, "Challenges for visual analytics," *Information Visualization*, vol. 8, no. 4, pp. 309–314, 2009.

[2] S. Delisle and B. Moulin, "User interfaces and help systems: from helplessness to intelligent assistance," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 117–157, Oct. 2002.

[3] D. C. Dryer, "Wizards, guides, and beyond: rational and empirical methods for selecting optimal intelligent user interface agents," in *Proc. 2nd Int. Conf. Intelligent User Interfaces*. New York, NY, USA: ACM, 1997, pp. 265–268.

[4] T. Selker, "COACH: a teaching agent that learns," *Communications of the ACM*, vol. 37, no. 7, pp. 92–99, Jul. 1994.

[5] J. Stocq and J. Vanderdonckt, "WOLD: a mixed-initiative wizard for producing multi-platform user interfaces," in *Proc. 9th Int. Conf. Intelligent User Interfaces*. New York, NY, USA: ACM, 2004, pp. 331–333.

[6] M. Wooldridge, "Intelligent agents: The key concepts," in *Proc. 9th ECCAI-ACAI/EASSS 2001, AEMAS 2001, HoloMAS 2001 on Multi-Agent-Systems and Applications II-Selected Revised Papers*. London, UK, UK: Springer-Verlag, 2002, pp. 3–43.

[7] C.-S. Lee, Y.-C. Chang, and M.-H. Wang, "Ontological recommendation multi-agent for tainan city travel," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6740–6753, Apr. 2009.

[8] M. Armentano, D. Godoy, and A. Amandi, "Personal assistants: Direct manipulation vs. mixed initiative interfaces," *Int. J. Human-Computer Studies*, vol. 64, no. 1, pp. 27–35, Jan. 2006.

[9] M. Frank, M. Muslea, J. Oh, S. Minton, and C. Knoblock, "An intelligent user interface for mixed-initiative multi-source travel planning," in *Proc. 6th Int. Conf. Intelligent User Interfaces*, ser. IUI '01. New York, NY, USA: ACM, 2001, pp. 85–86.

[10] T. M. Rhyne and A. MacEachren, "Visualizing geospatial data," in *ACM SIGGRAPH 2004 Course Notes*. New York, NY, USA: ACM, 2004. [Online]. Available: http://doi.acm.org/10.1145/1103900.1103931

[11] J. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Trans. on Graphics (TOG)*, vol. 5, no. 2, pp. 110–141, 1986.

[12] J. Mackinlay, P. Hanrahan, and C. Stolte, "Show me: Automatic presentation for visual analysis," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1137–1144, 2007.

[13] C. Stolte, D. Tang, and P. Hanrahan, "Polaris: A system for query, analysis, and visualization of multidimensional relational databases," *IEEE Trans. Visualization and Computer Graphics*, vol. 8, no. 1, pp. 52–65, 2002.

[14] O. Gilson, N. Silva, P. Grant, and M. Chen, "From web data to visualization via ontology mapping," in *Computer Graphics Forum*, vol. 27, no. 3. Wiley Online Library, 2008, pp. 959–966.

[15] C. Olston and J. D. Mackinlay, "Visualizing data with bounded uncertainty," in *Proc. IEEE Symp. Information Visualization*. Washington, DC, USA: IEEE Computer Society, 2002.

[16] T. Hengl and N. Toomanian, "Maps are not what they seem: representing uncertainty in soil-property maps," in *7th Inter. Symp. Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, M. Caetano and M. Painho, Eds., 2006, pp. 805+.

[17] S. Oyama, T. Kokubo, and T. Ishida, "Domain-specific web search with keyword spices," *IEEE Trans. Knowledge and Data Engineering*, vol. 16, pp. 17–27, 2004.

[18] L. Wanner, S. Vrochidis, S. Tonelli, J. Moßgraber, H. Bosch, Ari Karppinen et al , "Building an environmental information system for personalized content delivery," in *ISESS*, ser. IFIP Advances in Information and Communication Technology, J. Hrebícek, G. Schimak, and R. Denzer, Eds., vol. 359. Springer, 2011, pp. 169–176.

[19] M. Rospocher, "Pescado ontology documentation," Tech. Rep., 2010. [Online]. Available: http://www.pescado-project.eu/Pages/Pdfs-pages/PESCaDO_Ontology_Documentation_2.0.pdf

[20] E. Pianta and S. Tonelli, "KX: A flexible system for keyphrase extraction," in *Proc. of SemEval 2010*, 2010.

[21] The PESCaDO Project, "The PESCaDO Ontology." [Online]. Available: http://www.pescado-project.eu/ontology.php (retrieved 13. Sept. 2012).