Abstraction Levels as Support for UX Design of User's Interaction Logs

Juliana Jansen Ferreira(✉), Vinícius Segura, Ana Fucs, Rogerio de Paula, and Renato F.G. Cerqueira

IBM Research, Rio de Janeiro, RJ, Brazil
{jjansen,vboas,anafucs,ropaula,rcerq}@br.ibm.com

Abstract. User interaction logging is a powerful tool for user behavior studies, usability testing, and system metrics analysis. It may also be applied in large data contexts, such as social networks analysis, helping data scientists to understand social patterns. Data scientists, User Experience (UX) designers, Human-Computer Interaction (HCI) practitioners, and software engineers have been performing the analysis of this kind of data to obtain knowledge regarding the source system's usage. User interaction log data, however, can also be critical for final users themselves. They can use interaction log data, for example, (i) to revisit his own interaction path, redoing his steps that lead to a relevant insight or discovery; (ii) to learn from someone else's interaction path new ways to perform a given task; (iii) or even to analyze critical steps of a process supported by the source system. The need for final users to consume interaction log data is presenting significant challenges for UX researchers. Influenced by Semiotic Engineering, a HCI theory that views human-computer interaction as a form of human communication between designers and users mediated by a computer system, we propose three user interaction log abstraction levels - strategic, tactical, and operational - to frame and guide user interaction logs' UX design. In this paper, we discuss how those abstraction levels can be used as UX design guidelines and present some research questions to be explored - how source system captures interaction log is central for log analysis strategy and how a strategic level can be identified thought the analysis of interaction logs data from other abstraction levels.

Keywords: User interaction log data · UX design guidelines · Abstraction levels · Semiotic engineering

1 Introduction

User interaction logs are data input for different research approaches, used with different purposes, like user's behavior studies [1], usability testing [6,13], social network analysis [7] and web log analysis [3]. Logs are widely used by software developers to record valuable run-time information about software systems [17]. It is crucial to avoid logging too little or too much. To achieve so, developers

M. Antona and C. Stephanidis (Eds.): UAHCI 2017, Part I, LNCS 10277, pp. 369–382, 2017.
DOI: 10.1007/978-3-319-58706-6_30

need to make informed decisions on where to log and what to log in their logging practices during development [10]. The same "too little or too much" trade-off considerations apply to user interaction logs, considering the source system logged and the use of that log data.

Data scientists, User Experience (UX) designers, Human-Computer Interaction (HCI) practitioners, and software engineers have been performing the analysis of this kind of data to obtain knowledge regarding the source system's usage. User interaction log data, however, can also be critical for final users themselves. They can extract knowledge from previous interactions made by themselves, peers, or even a wider group. The final user can use interaction log data, for example, to revisit his own interaction path, redoing steps that lead to a relevant insight or discovery; to learn from someone else's interaction path new ways to perform a given task; or even to analyze critical steps of a process supported by the source system. The need for final users to consume interaction log data is presenting significant challenges for UX researchers.

Data volume challenges have been addressed by Visual Analytics (VA) research, investigating effective and efficient strategies to extract unknown and unexpected knowledge from data of unprecedentedly large size, high dimensionality, and complexity. VA approaches are based on the combination of data analysis and visualization techniques, which can handle this complex and dynamic data. They aim to explore the best interplay of computers' analytical capabilities and users' cognitive ones [4,15,16]. Once we have final users as consumers of interaction log data, the UX challenges need to be explored and combined with such VA strategies.

While interacting with a source system, the final user executes operations that are needed to carry out plans to achieve their final goals. When consuming the interaction log data, the final user's goals might have different focus and needs of data granularity and abstraction level. The UX challenge in this matter is to define the best abstraction level of the interaction log data to attend user goals regarding those data. The final user's goals towards interaction log data need to be well defined so the UX design to explore and use that data can be successful.

Approaches for analyzing human-computer interactions deal with sequential, integrated behavior rather than discrete actions [8, p. 75]. Hierarchical task analysis (HTA) [9, pp. 67–82] and Goals, Operators, Methods and Selection rules (GOMS) [14] are used to model and analyze interactions in a structured and composed way. Influenced by Semiotic Engineering [29], a HCI theory that views human-computer interaction as a form of human communication between designers and users mediated by a computer system, we propose three user interaction log abstraction levels – strategic, tactical, and operational – to frame and guide user interaction logs' UX design. Considering the log data from a source system, that data UX design in a operational level should communicate "how" the user executed some action, in a tactical level it should communicate "what" the user did while interacting with the source system and in a strategical level it should communicate "why" the user interacted with that system. Our users' interaction logs abstraction levels definition is a combination of two approaches: it considers the users' goals while exploring log data (top-down approach) and the available interaction data logged by a source system (bottom-up approach).

Considering those complementary perspectives, UX designers can identify how interaction log data can be used to enable users to take advantage of new knowledge generated by previous interaction with source systems. We use different examples of source systems to illustrate and discuss the user interaction logs abstraction levels.

In the end of this paper, we present some remarks and discussion points about the proposed abstraction levels to frame and guide user interaction logs UX design.

And present some research questions to be explored: how a source system captures interaction log is central for log analysis strategy and how a strategic level can be identified in analyzing interaction log data from other abstraction levels. We also leave some open questions for future research: if the source system nature (analytics-oriented or process-oriented systems) influences the interaction data log organization and presentation.

2 Multi-level Approaches for User Interaction

In HCI, multi-level approaches are commonly used to model and analyze user interactions. Hierarchical Task Analysis (HTA) [9, pp. 67–82] and Goals, Operators, Methods and Selection rules (GOMS) [14] are examples of those approaches. HTA [9, pp. 67–82] is a type of task analysis where a high-level task is decomposed into a hierarchy of sub-tasks. GOMS [14] involves the definition of a set of goals (what the user is trying to accomplish), operators (the elementary perceptual, motor or cognitive actions that are used to accomplish the goals), methods (for achieving the goals), and selections rules (for choosing among methods). These sets are used to model and analyze interactions in a structured and composed way. Those strategies help HCI practitioners and UX designers to breakdown users' interaction flow into minor actions to compose a contextualized interaction scenario.

Multi-level approaches are also used in other research areas to make sense of users' interaction. Roberts and colleagues [22] present an approach in visual analytics for decision-making, in which a combination of three provenance levels – data, analytical, and reasoning – supports the construction of rich narratives that encapsulate both explicit data and implicit knowledge. Their hypotheses are that visual analytics tools and methods can help to provide valuable means to make sense of these complex data. It can also help make tacit knowledge explicit and support the construction and presentation of the decision. Also from a visual analytics perspective, Segura and colleagues [24,26] propose a way to tell the user's interaction story using data logs as input. They present the interaction data in a segmented manner, indicating user's actions and system's actions. The user can look back on their exploratory interactions to, for example, recap a path to an insight.

3 History Visualization and the Consumption by the Final User

Many applications have an integrated log history, usually in the form of the undo/redo stack at the operational level. Visual analytics applications can present this history in multiple ways, usually integrating with the application's workflow. For example, sense.us [12] has a list of saved view images, whilst VisTrails [23,27] makes a graph of the operations that created a given visualization and allows users to return to a given node.

HistoryViewer [24,26] aims to represent the history from multiple applications (called source systems) at the tactical level, exploring the natural grouping of operational actions and the context of source system's domain. This makes it easier to final users consume the interaction history, since it decreases the amount of cognitive work to interpret the history, by reducing the number of events and describing them more related to the application actions then low-level input actions.

Between the many features of CATS (Cognitive Analytics Trail System) [30], it proposes the next operation to achieve a goal, suggests new tasks and goals, and allows the comparison of the recorded trails focusing on the processes. CATS relies on the users defining their current goals, operating at the strategical level.

Analyzing these applications, we can make a parallel to Schön's "reflections" [32, pp. 171–189]. Using the history whilst interacting with the application can be view as an example of *reflection in action*, since the user is thinking about his/her actions while performing them and influencing the interaction (*e.g.:* undo). Viewing the history after interacting with the application may be considered a *reflection on action*, since it happens after the interaction and is an opportunity to the user think about what s/he has done to perform the task. Finally, comparing the processes at a higher-level may be seen as a case of *reflection on practice*, analyzing the understanding around repetitive patterns.

4 Abstraction Levels in Log Analysis

The user's semiosis, humans' mental signification process, are associated with respect to the establishment of goals, to the plans he has devised to achieve his goals, or else to the operations that are needed to carry out plans and achieve goals [29]. We propose, therefore, using logs in multiple abstraction levels to frame and guide user interaction logs' UX design. We consider three abstraction levels:

- **Strategical:** establishment of goals;
- **Tactical:** plans/actions used to achieve a goal;
- **Operational:** operations performed to complete a plan.

These abstraction levels are hierarchical, *i.e.*, the strategic level may encompass several tactical ones, and the tactical level, several operational ones. The operational level deals with raw user's log data, typically local interactions (e.g.: user1 clicks in button "save"). These logs usually are domain-independent and can be extracted automatically. The tactical level adds a signification layer to user's log data, adding meaning spread over longer interactive paths. It requires more sophisticated semiotic and cognitive resources from users, who must re-signify their meaning according to the source system's domain (e.g.: user1 saves new report). Lastly, the strategic level is related to a broader context of user's plans performed while interacting with a source system. It adds even more signification to log data, also related to longer interactive paths and presenting knowledge about who the users are, what they want or need to do, how, and why (e.g.: manager John reports security incident). The strategic level is closely related to the user's goal, therefore existing only in the user's mental model. Moreover, it can involve multiple source system or even previous knowledge from the user.

We ground our proposal with the Semiotic Engineering theory [29]. Semiotic Engineering views HCI as a computer-mediated communication between designers and users that takes place during the interaction with the system. This communication is guided by the metacommunication template: "Here is my understanding of who you are, what I have learned you want or need to do, in which preferred ways, and why. This is the system that I have therefore designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision."

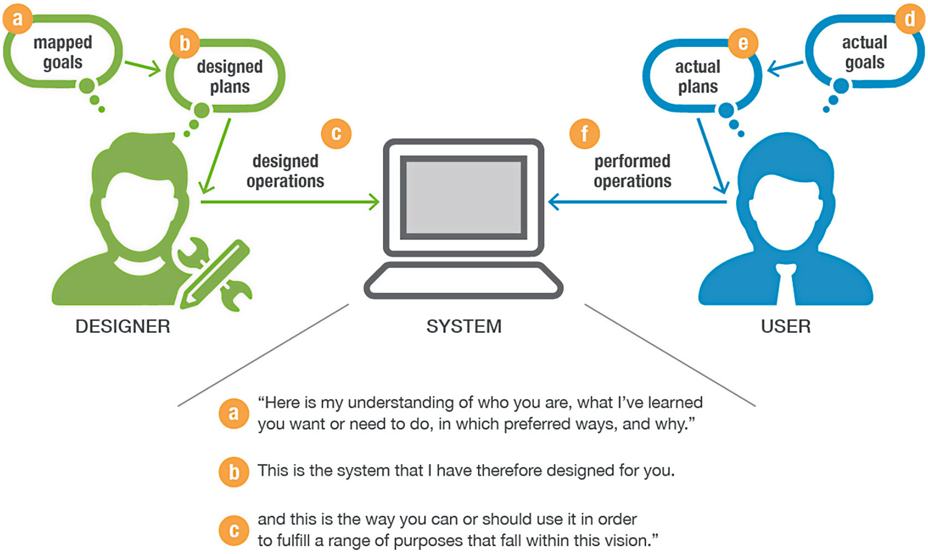

a
mapped goals
b
designed plans
c
designed operations
d
actual goals
e
actual plans
f
performed operations
DESIGNER
SYSTEM
USER
a
"Here is my understanding of who you are, what I've learned you want or need to do, in which preferred ways, and why."
b
This is the system that I have therefore designed for you.
c
and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision."

Fig. 1. Semiotic engineering.

Figure 1 illustrates this view. During design time, the designer must consider the user's goals within the system (a), associated with the "Here is my understanding of who you are, what I have learned you want or need to do, in which preferred ways, and why" part of the metacommunication template. From this understanding of user's goals, the designer develops a series of plans to support these goals (b), as expressed by the final part of the metacommunication template ("and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision"). Finally, the designer makes a set of operations available in the system (c) so the user may perform a designed plan – "This is the system that I have therefore designed for you."

During interaction time, the user establishes a goal (d), a plan to achieve it in the system (e), and performs a series of operations to complete the plan (f). Since the user's semiosis is unlimited, the designer's mental model (a/b), what assumed while designing the system, is not necessarily similar to the user's one (d/e). The distance between these mental models and, therefore, the message sent by the designer, may cause several communications breakdowns, at different abstraction levels [25,29]. The severity of the breakdown depends on the abstraction level [28]. Operational are usually local interaction problems, with little impact. Tactical breakdowns extends over longer interactive paths and may require more sophisticated semiotic and cognitive resources from users. Finally, strategical ones may be fatal, since it may signify a fundamental misconception regarding the user.

Our multi-level approach considers the source systems nature of users' interaction. We define a spectrum, as shown in Fig. 2, which goes from a process-oriented system, where users have to follow a sequence of action predefined by the system's designer, with little liberty to choose how to interact with that system; to a creativity-enabler system, where the user has total authoring control over his actions and produced artifacts. The systems only provides the resources to express that creativity. In that range are hybrid systems that combine different characteristics from both spectrum's edges. One example is a exploratory system, like visual analytics tools, where users can explore and play with a large amount of data to get insights and new knowledge about that data. This spectrum oriented discussions and reflections about our multi-level approach.

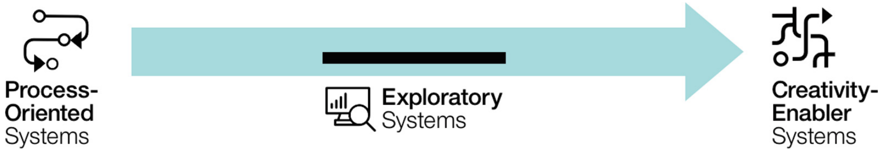


Fig. 2. Spectrum of systems nature of users' interaction.

In the following subsections, we present each of the abstraction levels of our proposal, with related works and examples.

4.1 Operational Level

The extraction of interaction log data is a rich source of information, providing knowledge concerning the user's trail when using a source system. Traditionally, user interaction log engines act at an operational level, which means capturing low-level interaction events, such as mouse movement, mouse clicks, and keyboard presses [5,33]. In mobile application, other type of events are obviously considered: tap, swipe, orientation change etc. Eye tracking techniques may also used to study the user's behavior by analyzing his eye movement data when interacting with a system [11,20]. Usually, timing information is also collected and remains useful when associated with contextual material. For example, using a think-aloud method where the user verbalizes about what he is doing when performing a task using an interface [18]. The audio record from a think-aloud usability study can be directly connected with the log timing data to allow a better understanding sequences of user actions sequences.

However, log and time stamp data by themselves can remain meaningless and generate an overflow of unstructured information. A straightforward strategy to start adding significance to log data at the operational level is to assign identification to interface objects. For instance, when dealing with web applications one can include extra tags to the HTML file in order to associate interaction events with the user's intention. This technique can help extracting meaning from each interaction event, indicating which button was clicked, which input text was filled, which UI element was selected etc.

The lack of a signification layer of operational logs creates a limitation on its analysis. It becomes an extremely hard task for the user to remind his mindset and understand the interaction events, by revisiting the steps appeared in log data during an interaction session. Considering the possibility of providing a storyline to the user for a late analysis of his interaction with a system, we must keep in mind that cross-related contextual data is essential.

There is a huge gap between interface manipulation (operational actions) and the user's goals or mindset (goals and mental model). The operation actions logs are the pieces that designers need to combine into a message to user about his previous interaction with a source system. Those pieces individually have little meaning, even to the user that executed the logged actions. That is why the UX design presented to revisit and explore that users' interaction log is a challenge. First, those pieces (operation actions) need to be assembled into plans (tactical) and later into goals (strategical).

4.2 Tactical Level

Looking only at the operational logging, we may understand *how* the user interacted with the system, but not *what* s/he was trying to accomplish. For example, in document editors, when the user activates the "bold" command, it may have different results: if there is a selected text, the current selection will be made bold and the text typed afterwards will be normal; if there is not a selected text, the text typed afterwards will be in bold. The operational logs will capture the interaction event, providing information if the “bold” command was activated by a keyboard shortcut, by the toolbar button, or the menu item. It will not, however, consider the usage context to differentiate between the results.

Whilst the goals’ mental model, and associated message (a), depends on the designer’s interpretation of the user’s goals (reinforced by the use of “here is my understanding” in the metacommunication template), the plans (b) are intrinsically connected with the available operations (a) (again, reinforce and by the use of “This is the system” in the metacommunication template). The designer, therefore, is aware of such plans and, consequentially, able to express logs in the tactical level. Going back to the example of the section’s beginning, the designer is aware of the different results of the “bold” command and may log according to the result – “formatted the selected text as bold” or “activated bold text”.

4.3 Strategical Level

The operational level focus on *how* the user interacts with the system and the tactical focus on *what* the user does. For example, in document editors, when the user activates the “bold” command, the operational logs will capture the interaction event, providing information if the “bold” command was activated by a keyboard shortcut, by the toolbar button, or the menu item. In the tactical level the designer can communicate the different results of the “bold” command and may log according to the result – “formatted the selected text as bold” or “activated bold text”. However, the *why* for the user to set that particular text in “bold” is registered on any log data. The source system could be instrumented to ask the user to communicate explicitly what every interaction goal is, but this is not viable for every move the user makes.

The strategical level adds another layer of context on log data, looking to point insights related to user’s goals, with his purposes while interacting. In the strategical level we aim to trail the reasoning and have knowledge about the *why*. In the metacommunication template (Fig. 1), its portion “and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision” (b) needs to be aligned with the user’s actual goal (d). This is a challenge by definition, since humans minds operates on a ongoing sense-making process, which can be influenced by anything in the environment, or previous knowledge, or the combination of both. Peirce calls it the unlimited semiosis [21, p. 231]. The semiosis is always changing, therefore, the user’s mental model regarding his goals is not a static concept.

In this dynamic setting of user’s goals, which are not loggable *per si*, we need a reference to what the user pretends when he uses the system. In this context, we go back to our definition of the spectrum of systems nature of users’ interaction presented in Fig. 2. Process-oriented system have a controlled interaction, normally guided by a process. For that class of system, we might have reference for users’ goals in the business processes related to the tasks performed in that system. Business process modeling [31] presents a set of structured knowledge that could be used to guide the UX design of logged data exploration, looking to provide support to users to identify their goals during a given interaction. The log of those processes instances, business process mining research explores that data [2], can also be reference for knowledge about user's goals.

Once we are able to collect knowledge about the strategical level, we can consider to provide some cognitive support to users about a goals while interacting with a source system, providing resources for inference about plans to achieve goals, suggestions of the next steps and so on (*e.g.* [30]). This is also related to the spectrum of systems nature of users' interaction (Fig. 2): the further the system is to the "process-oriented edge", the less any knowledge from previous interaction can help users to achieve their goals.

5 Examples of Multi-level Logging

In this section, we will present some examples of how the different abstraction levels could be used in different scenarios. We will start with an example of a chess game in Sect. 5.1, illustrating how the focus of the logged item can change according to the abstraction level. Section 5.2 presents an example of a document editor and how the use of abstraction levels can act as semantic grouping of log items. The next section (Sect. 5.3) illustrates a scenario of visual data exploration that may lead to unmapped goals. Finally, in Sect. 3, we present some applications features log visualization and compare their main goals with the different log abstraction levels being presented.

5.1 A Chess Game and the Change of Focus

We will start by considering a chess game log as seen in Table 1. At the operational level, we have the action of moving a piece from a coordinate to another one. This would be the lowest abstraction level, with little information regarding the context and the domain. If the system were instrumented, we can consider that it would be possible to know which chess piece was moved and to which position on the chess board. The focus would still be on the movement, with a little additional domain information ("Moved *white knight* from coordinates *G1* to *H3*")).

Table 1. Example of abstraction levels in a chess game.

Operational	**Tactical**	**Strategical**
Moved chess piece from coordinates *(x0,y0)* to *(x1,y1)* *or* Moved *white knight* from coordinates *G1* to *H3*	Attacked the *black bishop* at *H3* with the *white knight* from *G1*	Attacked the *black bishop* at *H3* with the *white knight* from *G1*, opening path to castling in the next turn

If we take a step further in the abstraction level, we can change the focus from the movement to the outcome of the movement. At the tactical level, we can focus that a chess piece was attacked with the movement, a terminology very specific to the chess game domain. Finally, at the strategical level, we can further explore the context and describe the wider context, for example, inferring a possible next movement ("castling" in our example).

5.2 A Document Editor and the Semantic Grouping

Another scenario is writing a paper in a document editor application. Given the objective of formatting a piece of text as a heading, a possible log can be seen in Table 2. At the operational level, we have the independent actions that the user executed, with little context from the system. At this level, we can differentiate between using the keyboard shortcut, the toolbar button, or the menu item.

Table 2. Example of abstraction levels in a document editor application.

Operational	Tactical	Strategical
Selected text "*Introduction*"	Formatted text "*Introduction*" as *bold* with *font size 14*	Formatted text "*Introduction*" as *heading 1*
Pressed "*Bold*" shortcut		
Selected "*font size*" *14*		

The tactical level groups a sequence of low level operations (bold and change font size commands) considering the context (the selected text). Given that a text was selected, The bold and change font size commands can be interpreted as formatting commands applied to the selected text, therefore, grouped into a single log item.

The strategical level can map the sequence of formatting commands to one of the available styles, associating the operations to a plan to achieve a goal. One possible influence in the UX design is informing the user about alternatives to achieve the same goal. In this scenario, the application could somehow indicate to the user that he could apply a pre-defined style, instead of repeating the individual formatting operations. This can make the user's plans mental model closer to the designer's one, making it an valuable learning strategy and avoiding future communication breakdowns.

5.3 A Visual Analytics Application and the Unmaped Goal Inference

Visual analytics applications stimulate data exploration from users. During such activity, the user's goal may even be outside of the application's scope, involving information from multiple sources, in different position in the Spectrum of systems nature of users' interaction (Fig. 2).

For example, WISE [19] enables user to view the distribution of a meteorological property (*e.g.:* temperature, precipitation) on a map, for a given forecast at a given timestep, as illustrated in Fig. 3.

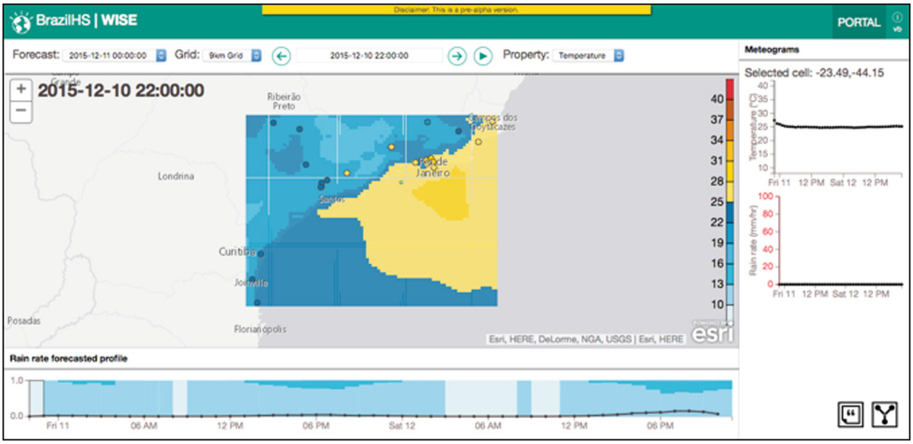
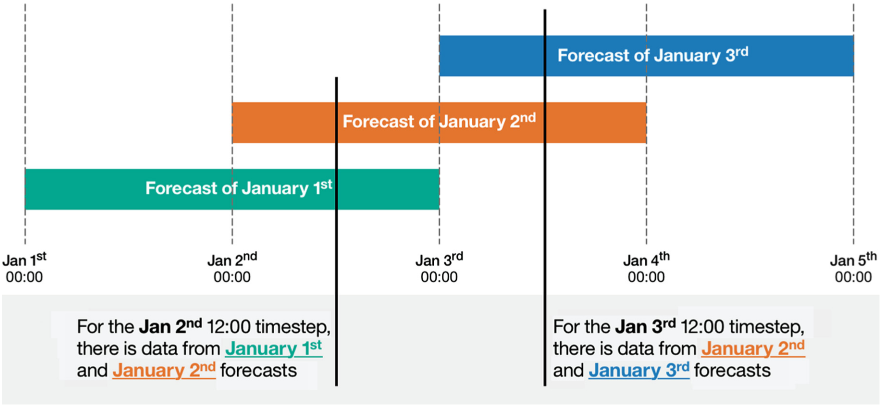


Fig. 3. WISE main UI, showing the map visualization.

A forecast is generated periodically (*e.g.:* daily) and has data for each timestep (*e.g.:* every hour) in a given time window (*e.g.:* 48 h). It is common that two different forecasts have an overlap, as examplified in Fig. 4.



For the **Jan 2nd** 12:00 timestep, there is data from January 1st and January 2nd forecasts

For the **Jan 3rd** 12:00 timestep, there is data from January 2nd and January 3rd forecasts

Fig. 4. Illustration of the forecasts overlapping.

It is possible, therefore, to have multiple forecasts covering the same timestep. This characteristic creates a usage scenario for WISE that was not originally mapped by its designer: compare the same timestep from different forecasts.

Table 3. Example of abstraction levels in WISE.

Operational	Tactical	Strategical
Selected forecast "*01/Jan*"	Viewed map at the *02/Jan 10am* timestep from the forecast generated on *01/Jan*	Compared the forecast for the *02/Jan 10am* timestep from the forecasts generated on *01/Jan* and *02/Jan*
Selected timestep "*02/Jan 10am*"		
Pressed "*Update*" button		
Selected forecast "*02/Jan*"	Viewed map at the *02/Jan 10am* timestep from the forecast generated on *02/Jan*	
Selected timestep "*02/Jan 10am*"		
Pressed "*Update*" button		

Table 3 shows an example of log generated by WISE[1]. At the operational level, we can see the individual operations available in the UI. The tactical level groups the operations, focusing on the moments when the map was updated.

The strategical level may infer the comparison goal from the series of tactical plans. This kind of analysis may be useful for the application's designer to discover new unmapped goals and what the users are doing to achieve them using the available operations.

6 Final Remarks

We presented our multi-level approach to frame and guide user interaction logs UX design, composed by three levels: operational, tactical, and strategical. By the examples and discussion presented, we have indications that the UX design of user interaction logs are directly related to the source systems logging strategy and instrumentation. Therefore, the source system's designers and developers are important reference of information for the UX designer of user interaction logs. They can also help to rebuild the "original message" of the source system, which can be used as a starting point for the UX design of user interaction logs.

We plan to use our abstraction level approach to discuss the UX design of domain specific system, which is placed in the middle of our spectrum. And we have plans to apply our abstraction level approach in system of different natures of users' interaction and see how can we improve that approach to address different source systems.

We have also identified some research questions to be explored in the future: (a) how a source system captures interaction log can influence the log analysis strategy and, (b) if a strategic level can be identified by analyzing interaction log data from other abstraction levels. We also leave a open question regarding our definition of a spectrum of systems nature of users' interaction: Does the source system nature (process-oriented to creatively-enabler systems) influence the interaction data log organization and presentation.

[1] To make the example simpler, we consider that there is an "Update" button that redraws the map given the current configuration. WISE actually redraws the map after each individual change in the configuration.

References

1. Understanding User Behavior Through Log Data and Analysis. New York, NY
2. Van der Aalst, W.M., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.: Workflow mining: a survey of issues and approaches. Data Knowl. Eng. **47**(2), 237–267 (2003)
3. Agosti, M., Crivellari, F., Di Nunzio, G.M.: Web log analysis: a review of a decade of studies about information acquisition, inspection and interpretation of user interaction. Data Mining Knowl. Discov. **24**(3), 663–696 (2012). http://dx.doi.org/10.1007/s10618-011-0228-8
4. Aigner, W., Bertone, A., Miksch, S., Tominski, C., Schumann, H.: Towards a conceptual framework for visual analytics of time and time-oriented data. In: Proceedings of Conference on Winter Simulation Conference, pp. 721–729 (2007)
5. Arroyo, E., Selker, T., Wei, W.: Usability tool for analysis of web designs using mouse tracks. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2006, pp. 484–489. ACM, New York (2006). http://doi.acm.org/10.1145/1125451.1125557
6. Atterer, R., Wnuk, M., Schmidt, A.: Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: Proceedings of the 15th International Conference on World Wide Web, pp. 203–212. ACM (2006)
7. Benevenuto, F., Rodrigues, T., Cha, M., Almeida, V.: Characterizing user behavior in online social networks. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, IMC 2009, pp. 49–62. ACM, New York (2009). http://doi.acm.org/10.1145/1644893.1644900
8. Carroll, J.M.: HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science. Morgan Kaufmann, Burlington (2003)
9. Diaper, D., Stanton, N.: The Handbook of Task Analysis for Human-Computer Interaction. CRC Press, Boca Raton (2003)
10. Fu, Q., Zhu, J., Hu, W., Lou, J.G., Ding, R., Lin, Q., Zhang, D., Xie, T.: Where do developers log? An empirical study on logging practices in industry. In: Companion Proceedings of the 36th International Conference on Software Engineering, pp. 24–33. ACM (2014)
11. Goldberg, J.H., Wichansky, A.M.: Eye tracking in usability evaluation: a practitioner's guide. Hyönä (2002, to appear)
12. Heer, J., Viégas, F.B., Wattenberg, M.: Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1029–1038. ACM (2007)
13. Hilbert, D.M., Redmiles, D.F.: Extracting usability information from user interface events. ACM Comput. Surv. (CSUR) **32**(4), 384–421 (2000)
14. John, B.E., Kieras, D.E.: Using GOMS for user interface design and evaluation: which technique? ACM Trans. Comput.-Hum. Interaction (TOCHI) **3**(4), 287–319 (1996)
15. Keim, D.A., Mansmann, F., Oelke, D., Ziegler, H.: Visual analytics: combining automated discovery with interactive visualizations. In: Proceedings of the 11th International Conference on Discovery Science, DS 2008, pp. 2–14 (2008)
16. Kohlhammer, J., Keim, D., Pohl, M., Santucci, G., Andrienko, G.: Solving problems with visual analytics. Procedia Comput. Sci. 7(0), 117–120 (2011). http://www.sciencedirect.com/science/article/pii/S1877050911007009. Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 2011)

17. Li, H., Shang, W., Hassan, A.E.: Which log level should developers choose for a new logging statement? Empirical Softw. Eng. 1–33 (2016)
18. Nielsen, J.: Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco (1995)
19. Oliveira, I., Segura, V., Nery, M., Mantripragada, K., Ramirez, J.P., Cerqueira, R.: WISE: a web environment for visualization and insights on weather data. In: WVIS - 5th Workshop on Visual Analytics, Information Visualization and Scientific Visualization, SIBGRAPI 2014, pp. 4–7 (2014). http://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/11954/WVIS-SIBGRAPI-2014.pdf?sequence=1
20. Pan, B., Hembrooke, H.A., Gay, G.K., Granka, L.A., Feusner, M.K., Newman, J.K.: The determinants of web page viewing behavior: an eye-tracking study. In: Proceedings of the 2004 Symposium on Eye Tracking Research and Applications, ETRA 2004, pp. 147–154. ACM, New York (2004). http://doi.acm.org/10.1145/968363.968391
21. Peirce, C.S., Houser, N.: The Essential Peirce: Selected Philosophical Writings, vol. 2. Indiana University Press, Bloomington (1998)
22. Roberts, J., Keim, D., Hanratty, T., Rowlingson, R., Walker, R., Hall, M., Jacobson, Z., Lavigne, V., Rooney, C., Varga, M.: From ill-defined problems to informed decisions. In: EuroVis Workshop on Visual Analytics (2014)
23. Santos, E., Lins, L., Ahrens, J.P., Freire, J., Silva, C.T.: VisMashup: streamlining the creation of custom visualization applications. IEEE Trans. Vis. Comput. Graph. **15**(6), 1539–1546 (2009)
24. Segura, V., Ferreira, J.J., Cerqueira, R.S.B.: Uma avaliação analítica de um sistema de visualização do histórico de interação do usuário usando cdn e pon. In: Brazilian Symposium on Human Factors in Computing Systems 2016 (2016)
25. Segura, V., Simões, F., Sotero, G., Barbosa, S.D.J.: Multi-level communicability evaluation of a prototyping tool. In: Kurosu, M. (ed.) HCI 2013. LNCS, vol. 8004, pp. 460–469. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39232-0_50
26. Segura, V.C.V.B., Barbosa, S.D.J.: History viewer: displaying user interaction history in visual analytics applications. In: Kurosu, M. (ed.) HCI 2016. LNCS, vol. 9733, pp. 223–233. Springer, Cham (2016). doi:10.1007/978-3-319-39513-5_21
27. Silva, C.T., Anderson, E., Santos, E., Freire, J.: Using VisTrails and provenance for teaching scientific visualization. In: Computer Graphics Forum, vol. 30, pp. 75–84. Wiley Online Library (2011)
28. de Souza, C., Leitão, C.F.: Semiotic Engineering Methods for Scientific Research in HCI. Morgan and Claypool Publishers, San Rafael (2009)
29. de Souza, C.S.: The Semiotic Engineering of Human-Computer Interaction (Acting with Technology). The MIT Press, Cambridge (2005)
30. Thiago, R., Azevedo, L., da Silva, V., Segura, V., dos Santos, M., Cerqueira, R.: CATS: cognitive analytic trail system (2016). http://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12587
31. Van Der Aalst, W., Van Hee, K.M.: Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge (2004)
32. Winograd, T.: Reflective conversation with materials an interview with Donald Schön by John Bennett. In: Bringing Design to Software, 1st edn., pp. 171–189. Addison Wesley, Harlow (1996)
33. Zheng, N., Paloski, A., Wang, H.: An efficient user verification system via mouse movements. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, pp. 139–150. ACM, New York (2011). http://doi.acm.org/10.1145/2046707.2046725