# A Visual Approach towards Knowledge Engineering and Understanding How Students Learn in Complex Environments

**Lauren Fratamico**
University of British Columbia
Vancouver, Canada
fratamic@cs.ubc.ca

**Sarah Perez**
University of British Columbia
Vancouver, Canada
sarah.perez@ubc.ca

**Ido Roll**
University of British Columbia
Vancouver, Canada
ido.roll@ubc.ca

## ABSTRACT

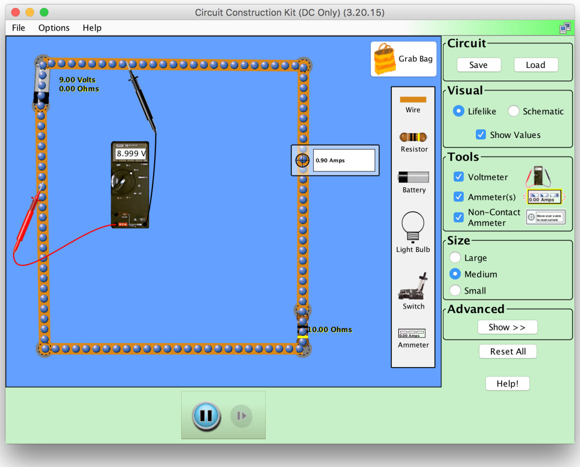
Exploratory learning environments, such as virtual labs, support divergent learning pathways. However, due to their complexity, building computational models of learning is challenging as it is difficult to identify features that (i) are informative with respect to common learning strategies, (ii) abstract similar actions beyond surface differences, and (iii) differentiate groups of learners. In this paper, we present a visualization tool that addresses these challenges by facilitating a novel analytic approach to aid in the knowledge engineering process, focusing on five main capabilities: data-driven hypotheses raising, visualizing behavior over time, easily grouping related actions, contrasting learners' behaviors on these actions, and comparing the behaviors of groups of learners. We apply this analytic approach to better understand how students work with a popular interactive physics virtual lab. By splitting learners by learning gains, we found that productive learners performed more active testing and adapted more quickly to the task at hand by focusing on more relevant testing instruments. Implications for online virtual labs and a broader class of complex learning environments are discussed throughout.

## ACM Classification Keywords

H.5.3. Information Interfaces and Presentation (e.g. HCI): Group and Organization Interfaces; K.3.1. Computers and Education: Computer Uses in Education

## Author Keywords

Visual analytics; Exploratory data analysis; Interactive virtual labs; Exploratory learning environments; Learning strategies; Temporal data; Educational data mining; Learning analytics



Visual: ● Lifelike ○ Schematic
☑ Show Values
Tools: ☑ Voltmeter ☑ Ammeter(s) ☑ Non-Contact Ammeter
Size: ○ Large ● Medium ○ Small

**Figure 1. The PhET CCK interface, an unstructured learning environment. A complete circuit has been built in the workspace with a battery, resistor, and some wires. Also seen are two of the testing instruments: voltmeter (measuring voltage) and ammeter (measuring current).**

## INTRODUCTION

Online learning environments are increasingly complex. One aspect of this complexity is the diversity of instructional activities and their affordances. In addition, many environments move away from prescribed linear trajectories to support (and encourage) user-driven exploration. Another aspect is the variety of learners, their goals, and engagement patterns. Overall, this complexity and diversity add a significant challenge to the interpretation of log data from learning trajectories. In short, how can we interpret learners' interactions in online learning environments? More specifically, how can we identify strategies, infer intentions, assess learning, or evaluate quality of engagement? For example, Figure 1 shows a virtual lab in which learners construct and test electric circuits with the goal of learning about DC circuits. This virtual lab has 124 different actions available to learners, and is part of a family of virtual labs that are used over 50 million times a year [1]. How can one use log data to evaluate student's attitudes and knowledge when the design space is unlimited and the solution space is underdefined? Furthermore, how can this be applicable to learners with diverse backgrounds and goals?

To interpret learner behaviors, researchers create models of learning in the environment. Common approaches for knowledge engineering rely on expert analysis and learning theories. However, all too often, this process is not well-informed by

*L@S 2017,* April 20-21, 2017, Cambridge, MA, USA

DOI: http://dx.doi.org/10.1145/3051457.3051468

empirical data. Machine learning approaches, on the other hand, make extensive use of empirical data, yet often ignore expert knowledge and learning theories. In this paper we seek to combine the benefits of theory-driven (top down) knowledge engineering and data-driven (bottom up) knowledge discovery to make sense of data from complex systems. We propose a workflow, supported by a system, which allows researchers to hypothesize patterns based on data and quickly test these hypotheses. By doing so, we infuse the knowledge discovery process with insights based on available data.

A main goal for this work is to strengthen the bridge between data sciences, learning sciences, and instructional design. The tool, Tempr, helps its users raise hypotheses about what actions are related and how. It then allows them to quickly test their hypotheses by visualizing the relationship between engagement behaviors and other student-level factors (such as knowledge level). Tempr facilitates an analytic process that is based on the following approaches: iterative hypothesis raising and testing regarding grouping of related actions, informed by data; temporal analysis of log data; and visual comparisons between groups of learners.

We begin by describing the challenge of knowledge engineering in complex environments. We then review related literature from the education and visualization communities. We highlight key capabilities of Tempr that address many of the identified challenges, and demonstrate its utility by presenting the results of using Tempr to analyze authentic data from the virtual lab shown in Figure 1. Last, we describe how Tempr could assist with knowledge engineering in other complex learning environments, such as Massive Open Online Courses (MOOCs).

## BACKGROUND

A first step towards modeling learning in new environments is to better understand it. That is, researchers should be able to label patterns in the data and assign meaning to sequences of actions. Overall, there are two common methodologies for contextualizing our understanding of what learning looks like in a target environment. While these typically apply to model the domain knowledge, they can also be applied to modeling the learning process [19]. Knowledge engineering refers to a theory-driven top-down process, often based on expert analysis [2]. For example, in a problem solving environment, fast repeat failed attempts can be labeled as guessing. This label is based on our understanding of how students learn, not on empirical data. Similarly, knowledge engineering can be used successfully to interpret other aspects of the learning process, such as help seeking [2, 18]. However, knowledge engineering becomes harder as the design space in the environment grows. For example, in a virtual lab such as the one shown in Figure 1, it is nearly impossible to define what actions constitute different testing strategies, given the multitude of testing actions. Similarly, in a MOOC setting, interpreting navigation events is challenging given their diversity. A common alternative to knowledge engineering is knowledge discovery, in which machine learning and statistical approaches are used to extract patterns from the data. For example, a knowledge-discovery approach to identify guessing in a problem-solving environment revealed two different types of guessing [4]. Knowledge discovery has been found to be very successful in certain domains [10], including in complex environments [3, 8]. However, while knowledge discovery can often be applied to predict overall learning from an environment, it has several major shortcoming. First, it is often hard to interpret the detected trends, and thus, while accurate, does not inform theory [2, 19]. Second, it is effective for skills that are easy to label, but less for divergent strategies [3, 21]. Last, the detected models may be overly specific to context and populations [8].

Several existing visualization systems address similar challenges to the ones identified above. A few systems support temporal analysis of log data and visual comparisons between groups [12, 14, 16, 24]. A number of these systems allow for iterative hypothesis raising and testing regarding grouping of related actions. One such system is CoCo [14], which allows users to test their hypotheses about which events, or sequences of events, describe different clusters of learners. However, CoCo's design works best when investigating log data with a small number of possible events. Thus, applying the tool to data from environments with a large variety of actions is challenging. Similarly, MatrixWave [24] allows for the comparison of log events between two groups. Specifically, it allows for the exploration of how one webpage is navigated in two different conditions. This allows researchers to form hypotheses regarding the differences between the two conditions. However, again, this tool works best when researchers have a small set of events to study, it does not allow the researcher to combine multiple events together to explore how different types of events relate to each other.

Other visualization systems have made an attempt at explicitly highlighting informative features from the data. For example, INFUSE [11] visualizes the result of feature selection, showing the user the most predictive features. While this is an important step, the system does not support the next steps, in which analysis is preformed on grouped events. FeatureInsights [6] helps users to engineer features from their text data, highlighting the features that distinguish two groups. However, this system also does not allow users to group multiple features. Others allow for detection and automated grouping of similar features [9, 23], an approach more similar to knowledge discovery. Yet, we advocate for a process in which the researcher can form their own groups of features as is desired in feature engineering. Other researchers attempted to group log event features for their visualization of clickstream data in semantic ways, including tf-idf and LDA, but resorted to relying on experts to perform the grouping manually, as none of the automated groupings produced acceptable results [13]. Thus, while addressing the challenge of making meaning of complex data is shared in many existing environments, it seems that these often lack the ability for the user to group different types of actions, evaluate that grouping over time, and analyze how the newly-formed groups are used by different clusters of learners. Tempr and the analytic workflow that it facilitates address that need.

## VISUALIZATION TOOL

We have developed Tempr, a visual analytic approach, to assist with knowledge engineering. The main goal of this tool is to inform the top-down knowledge engineering process with patterns that emerge bottom up. Tempr does so iteratively: researchers investigate different types of behaviors and use the tool to identify patterns in the data and informative groupings of actions. Assigning meaning to these patterns can facilitate additional hypothesis testing and grouping in Tempr. The outputs of this exploratory process are intended to inform additional analysis, likely more statistical in nature.

### Capabilities

The tool is built to support five main capabilities, presented below.

*Surfacing big picture patterns*

It is important to give researchers an overview of the data upfront, so that they may better discover and interpret emerging patterns. A global view of the data allows researchers to assess the scope and scale of the data [22], in terms of, for example, the number of available actions in a virtual lab or other learning environment. Acknowledging the size and scale of a dataset will help put discovered patterns in perspective. Finally, an overview of the data allows researchers to prioritize the aspects or features of the data they wish to explore.

*Visualizing learning over time*

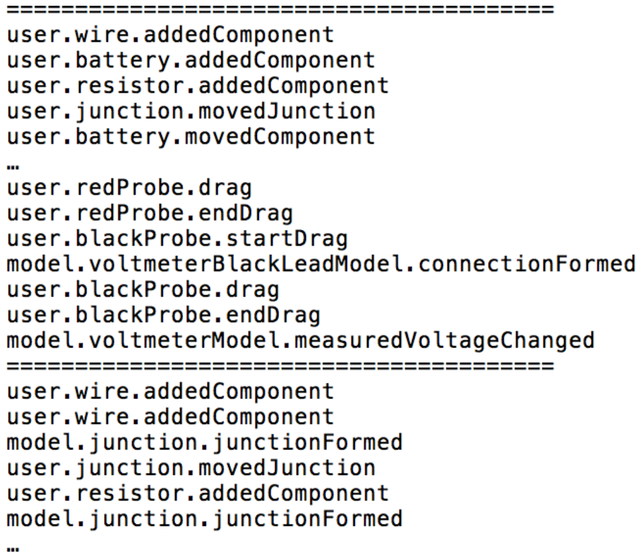
While researchers often look at data by learner, too often data across entire sessions (or even entire courses) are combined, ignoring trends over time. However, learning in complex environments has a temporal nature [7]. For example, while an early pause in a student's actions may be a sign of planning, a later pause is more likely a sign of reflection. Analyzing a student's overall pausing behavior ignores possible strategic use of pauses over time. Thus, Tempr visualizes how learning unfolds over the duration of the activity.

*Supporting exploratory grouping of related actions*

One of the main challenges in exploratory environments, as described above, is the large diversity of actions. These could be merged into fewer types of actions, skills, or strategies. However, identifying which actions should be merged together in order to abstract learner behaviors is not a trivial task. As an example, in the case of a MOOC, correct problem-solving attempts may show similar trends to incorrect ones, and may be grouped to create an "attempt" category. However, perhaps these actions are qualitatively different, as they hint at different knowledge levels and entail different subsequent actions. Thus, quick evaluation of grouping of actions is of interest. Similarly, in an interactive virtual lab such as the one shown in Figure 1, grouping and then visualizing all "building" types of actions may be more meaningful than individually visualizing actions that are associated with building a circuit, such as adding a wire, connecting a light bulb, or connecting a resistor. Tempr supports quick evaluation of potential grouping of actions to facilitate the evaluation of learning strategies.

*Contrasting of actions*

While some actions need to be grouped, other types of actions need to be contrasted. For example, in a MOOC context, one may wish to compare how learners engage differently with graded versus ungraded problems, quizzes in different course modules, lecture videos in different course modules, etc. In a virtual lab, one may wish to understand what the relationship is between building circuits and testing them. The tool supports visual contrasts by simultaneously displaying the distribution of relevant actions over time, and, in so doing, enables researchers to evaluate these actions as part of effective or ineffective learning strategies.



```
========================================
user.wire.addedComponent
user.battery.addedComponent
user.resistor.addedComponent
user.junction.movedJunction
user.battery.movedComponent
…
user.redProbe.drag
user.redProbe.endDrag
user.blackProbe.startDrag
model.voltmeterBlackLeadModel.connectionFormed
user.blackProbe.drag
user.blackProbe.endDrag
model.voltmeterModel.measuredVoltageChanged
========================================
user.wire.addedComponent
user.wire.addedComponent
model.junction.junctionFormed
user.junction.movedJunction
user.resistor.addedComponent
model.junction.junctionFormed
…
```

**Figure 2. Input to Tempr. All users for each group are placed in one ".txt" file. Each action they took while working with the environment is listed one after the other, with users separated by "======".**

*Comparing groups of learners*

In many exploratory environments there is no one correct way to engage with the course. For example, how should learners make use of self-tests in MOOCs? How should they use testing instruments in a virtual lab? Tempr has a built-in ability to show how different groups of learners interact with the environment differently. This could be used to compare learners with different outcome attributes (such as comparing successful learners to less successful ones), or incoming attributes (such as comparing the behavior of learners with different prior knowledge, attitudes, or backgrounds).

### Tempr's architecture

The main Tempr interface can be seen in Figure 4. It was built using JavaScript[17], and, specifically, D3.js [5] for the graphs. Tempr is available for download on Github under the GNU license: https://github.com/fratamico/Tempr---A-visual-knowledge-engineering-tool.

### Data Input

Tempr was designed with flexibility and generality in mind. The requirements in terms of data input reflect this design by enabling the use of log data from diverse sources with varying analysis goals. Tempr takes input in the form of two

".txt" files, one for each group of learners. Each file is comprised of the sequence of log events for each user in that group. An example of the input is shown in Figure 2. In terms of format, users' action sequences are separated by equals signs and each line is a logged event for that user. The researcher can choose which and how many arguments within a logged event they want to include, and should separate the selected arguments with a period. For example, in the virtual lab shown above, we chose to include the following information in the event: actor (user or model), component (the component being operated on), and action (the action that was done). For MOOCs, this data can include module ID, component type, component ID, and event type, for example: Week1.video.Introduction_Video.PlayVideo. There are no requirements of event names. However, the terms in each line of the logged events can later be used for filtering and should include all meaningful contextual information available in the logged event. Notably, the input data files contain sequences of events and do not include duration information. While this compromises the ability to analyze actions by duration, it makes data entry to Tempr more straightforward and permissible to log data without time stamps.

As mentioned above, Tempr takes two files, one for each group of learners. Learners can be grouped in different ways. In the example below we split learners into two groups based on learning gains from an activity in a virtual lab. However, learners could be divided in other ways too, for example, based on incoming attributes (knowledge, attitudes) or whether they completed the MOOC or not. Any qualitative or quantitative factor can be used to split students into groups and these groups need not be of the same size.

## An overview of Tempr

Tempr has three main panels:

1. The Heatmap Panel. This panel supports hypothesis raising. It provides an overview of all actions in the tool over time and helps to identify which actions show similar patterns and could be further investigated.
2. The Merging Panel. This panel supports exploration of different groupings of actions. This is key for knowledge engineering, as it helps to see what combined actions may highlight the differences in how groups of students learn.
3. The Visualization Panel. This panel enables the comparison of an action or merged sets of actions by groups of students over time.

The next sections present the utility of these panels by demonstrating how users can quickly raise hypotheses, identify and design the comparisons they wish to make, discover differences in learning behaviors of student groups, and test their hypotheses.

## Heatmap Panel

The heatmap panel offers users the big picture of their data. It visualizes differences in the frequency of action use at certain time intervals. This helps users raise hypotheses about learning in their environment. For example, users may be curious about which events are similar between different groups of learners. Similarly, the heatmap allows the comparisons of actions based on the use of the action by student groups. Finally, potential groupings of actions can be determined by identifying actions sharing similar use patterns over time.

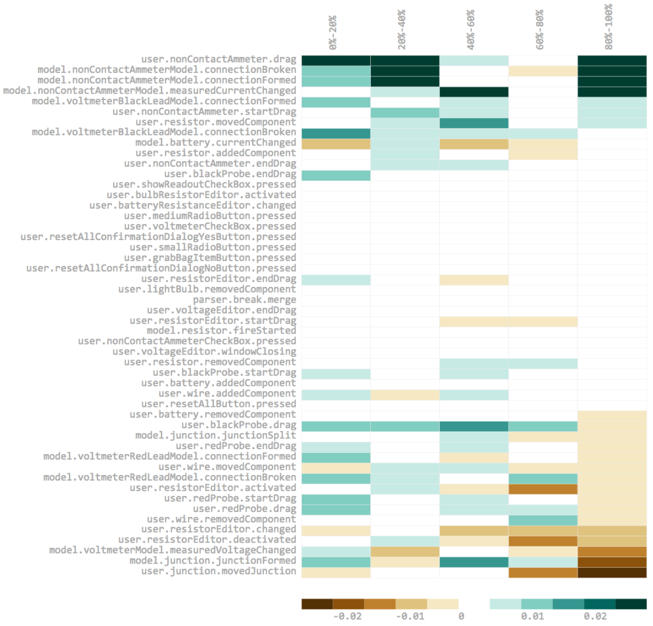


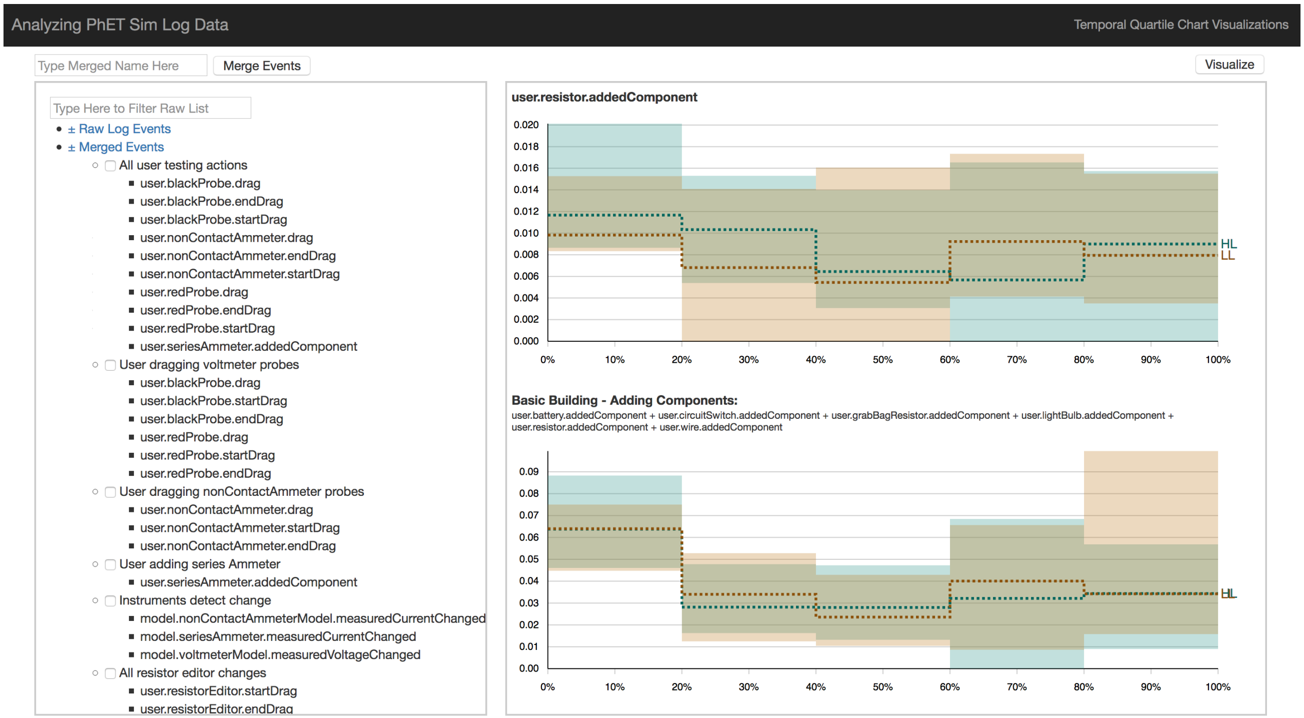
**Figure 3. The heatmap panel of the Tempr interface. Blue hues indicate that high learners (HL) performed the action more, and brown hues indicate that low learners (LL) performed the action more. The darkness of the color indicates how much more that group performed the action.**

The heatmap panel can be seen in Figure 3. Along the left are the raw log events. Raw log events are simply each event that is available in the logs of raw student action data. For virtual lab log data, one of these such events could be the action of manipulating a testing instrument. For MOOC data, this could be a play of video 1 in lesson 1. Across the top shows the percent of actions completed, normalized for each student. For example, the left column, labeled "0%-20%", summarizes the first 20% of actions that students took. The colors indicate which group of students performed the action more, and the darkness of the color indicates the how much more of it they performed.

This panel gives an overview of how groups of learners were performing over the course of interaction. We can quickly ignore the events where the heatmap shows white cells, as these are times when that event was used with a similar frequency by the two groups of learners. This allows users to focus on the more divergent events and form hypothesis about which actions should be combined to distinguish groups of learners while also developing potentially informative features.

## Merging Panel

The merging panel allows users to group actions that they hypothesize are related. As mentioned in the introduction, in order to find patterns in data, a researcher may have to try looking at the data in a variety of different ways. To determine



- ± Raw Log Events
- ± Merged Events
  - ○ □ All user testing actions
    - user.blackProbe.drag
    - user.blackProbe.endDrag
    - user.blackProbe.startDrag
    - user.nonContactAmmeter.drag
    - user.nonContactAmmeter.endDrag
    - user.nonContactAmmeter.startDrag
    - user.redProbe.drag
    - user.redProbe.endDrag
    - user.redProbe.startDrag
    - user.seriesAmmeter.addedComponent
  - ○ □ User dragging voltmeter probes
    - user.blackProbe.drag
    - user.blackProbe.startDrag
    - user.blackProbe.endDrag
    - user.redProbe.drag
    - user.redProbe.startDrag
    - user.redProbe.endDrag
  - ○ □ User dragging nonContactAmmeter probes
    - user.nonContactAmmeter.drag
    - user.nonContactAmmeter.startDrag
    - user.nonContactAmmeter.endDrag
  - ○ □ User adding series Ammeter
    - user.seriesAmmeter.addedComponent
  - ○ □ Instruments detect change
    - model.nonContactAmmeterModel.measuredCurrentChanged
    - model.seriesAmmeter.measuredCurrentChanged
    - model.voltmeterModel.measuredVoltageChanged
  - ○ □ All resistor editor changes
    - user.resistorEditor.startDrag
    - user.resistorEditor.endDrag

user.battery.addedComponent + user.circuitSwitch.addedComponent + user.grabBagResistor.addedComponent + user.lightBulb.addedComponent + user.resistor.addedComponent + user.wire.addedComponent

**Figure 4. Tempr interface. The merging panel is on the left, and the visualization panel is on the right. The top graph shows the use of one action over time, specifically the use of the action "user.resistor.addComponent". The bottom graph shows the use of a group of merged actions over time, specifically the use of all actions that result in a circuit component being added, such as a light bulb, resistor, battery, wire, etc.**

the best features to engineer, one may first want to abstract to a bigger picture (combining many raw log events), then dive in and explore different pieces that make up the bigger picture, then abstract out to a different bigger picture, etc. This iterative process is at the core of the Tempr workflow.

There are two portions of the merging panel which can both be seen on the left side of Figure 4: the raw log events portion and the merged events portion. There could be hundreds of different types of raw log events, as is often common in exploratory learning environments. This drives the need to combine raw events to understand how sets of actions are applied by different groups of learners throughout the interaction. Merged events are comprised of multiple raw log events. To merge raw log events together with Tempr, a researcher can first select the raw log events in the top of the left panel by checking the box next to them, then merge them by clicking on the "Merge Events" button above. Subsequently, the merged action will appear under the "Merged Events" list and all the raw log events that comprise it will be listed below it. This feature allows for an iterative procedure for hypothesis testing. Users can quickly group actions to test hypotheses about what impacts different groups of combined raw log actions will have on learner behavior, and revise accordingly, combining bottom-up and top-down analytic processes: expert knowledge guides the data mining, by choosing which combinations to test; patterns in the data then guide the expert knowledge, by revealing which show consistent and coherent terms.

### Visualization Panel

The visualization supports comparing different groups of learners on different sets of actions over time. Both merged events and raw log events (top graph, right side) can be plotted, as shown in the right hand side of Figure 4. The bolded title gives the name of what is graphed, and the subtitle below, if any, tells what raw events were combined to make up the graph. The bottom graph in Figure 4 is the visualization of the "Basic Building - Adding Components" set. To comprise this feature, 6 raw log events, all related to adding components to the learning environment, were merged. In this way, Tempr allows a researcher to gain a visual understanding of how different combinations of raw events reveal learning patterns for different groups of students.

Each chart is an overlay of two plots, corresponding to the two groups of learners. The y-axis is frequency of that action, that is, the percent of this action out of all actions per student during that time slice, averaged across students. For example, if a student performed 20 actions while interacting with the environment, and 2 of them were testing actions, then the frequency for that student would be 0.1. The presented value is the average frequency across students in that group. The x-axis across the bottom shows percent of actions completed, normalized for each student. In this way, with Tempr, users can compare student performance over the learning process and see how student groups perform differently over the course of interacting with the learning environment. The visualization over task progression is an essential piece as it helps us understand how learning unfolds over time. Furthermore, differences in frequency of actions taken can often be washed out if only looking over the overall interaction patterns over the entire activity. Visualizing task progression allows users to see things such as changes in frequency for each student group over time, differences between the two groups of learners, and frequency of sets of actions in specific time slices relative to frequency of other actions. For example, in Figure 4, we can see from the bottom graph that adding components to a circuit in the PhET virtual lab is done more frequently at the beginning of interaction than it is at any other point throughout interaction.

The graph itself shows distribution within groups. The dashed lines are the median frequency of that action for users in each group, and the shaded regions shade the region between the 25th and 75th percentile for each group (one may think about it as box-and-whisker plots for each group at each time slot, without the whiskers). This representation shows both central tendency and distribution, and is less sensitive to outliers. The two groups can be quickly identified based on the colors. In this way, users can quickly compare the frequency with which most users in each group are performing each action. There is also a green area in the middle where the two groups of learners are overlapping. This is common, and it can be common for the green area to be large; for the most part, but depending how students were grouped, many students act fairly similarly throughout interacting with the environment. That's one of the challenges in finding the right features that differentiate between groups of learners. However, with Tempr, since the medians are also graphed and since the area between the quartiles can extend past each other (eg, the 75th percentile for one group may be higher than it is for the other group, as it is in the first 20% of interaction in the top graph in Figure 4), we can still easily understand these subtle differences in how the groups of students are learning. It is important to emphasize that data sets with large and statistically significant effect sizes still have much overlap in their distributions. Overall, the visualization panel of Tempr allows for easy comparison for how different groups of students are performing.

## RESULTS

In this section we discuss the application of Tempr to facilitate analysis on students working through the PhET CCK learning environment. We first describe the PhET CCK and its dataset. We then demonstrate how Tempr is able to abstract and evaluate learner strategies from user log data.

### CCK Simulation

PhET is a family of over a hundred interactive virtual labs in STEM topics, used by students at the kindergarten through university level [1]. It is the most popular family of virtual labs, having over 50 million runs a year. These virtual labs offer learners opportunities to engage in authentic inquiry, and teachers create a variety of activities around these. These are often shared in the teaching community.

The PhET Circuit Construction Kit (CCK) is the most commonly used virtual lab in the PhET family. Students in CCK construct and test DC electric circuits by using a variety of components that include batteries, wires, light bulbs, resistors, and measurement instruments such as ammeters and voltmeters. Overall, there are 124 different types of actions that students can perform at each moment. These actions include adding, moving, joining, splitting, and removing components, as well as changing the attributes of components (such as resistance). Additional actions relate to the interface (such as changing views or zooming in and out), or the virtual lab itself (such as pausing or resetting the virtual lab). The outcomes of these actions depend on the state of the virtual lab. These outcomes manifest themselves in the logs in the form of model actions. For example, the user action of changing the resistance of a component will, if that component is connected to a live circuit, trigger a model action of changing the current of the circuit. A second model action in this scenario may be a change to the ammeter reading, if one is connected to the circuit.

#### *User Study*

One hundred students from first-year physics courses at the University of British Columbia volunteered for a study which took place outside their normal classroom hours [20]. The study included an activity on the topic of DC circuits, which took 30 minutes. The activity asked students to explain the effect of connecting multiple resistors on the voltage and current of a circuit. Students received this general learning goal and a general recommendation to explore several resistors within the same circuit loop, on different circuit loops, and a combination of the two. Pre- and post- tests were given to each student so that we could measure learning gains across the activity.

#### *Processing the Log Data*

To prepare the data, we extracted the sequential list of actions logged as each student interacted with the virtual lab. As mentioned above, this was a combination of student-originated actions and the resulting model actions. Classifying learners to two groups was done by calculating their learning gains [15]. We then applied a tertiary split, comparing the high learning gain group (HL) to the students with low learning gains (LL), ignoring the middle third.

### CCK Simulation Analysis with Tempr

Overall, our analysis was driven by the following question: How do students learn by testing? Without Tempr, one may settle for merging all testing events, looking for testing more vs. less frequently during the entire activity. However, as shown in Figure 5, this does not reveal interesting results: HL test slightly more than LL at the beginning and end of interaction, but during the rest of interaction, it appears they test fairly similarly. Instead, we want to better understand how testing instruments are used by learners of both groups with the goal of identifying effective and ineffective testing strategies. Specifically, we looked to better understand: (i) How do learners use different testing instruments? For example, are they all treated equally, or do different groups gravitate to different instruments? (ii) How do learners use the instruments? Do they leave them connected to passively test other changes in the circuit, or do they test actively by moving them around? Do we observe the same patterns for measuring voltage and measuring current? Answering these questions can

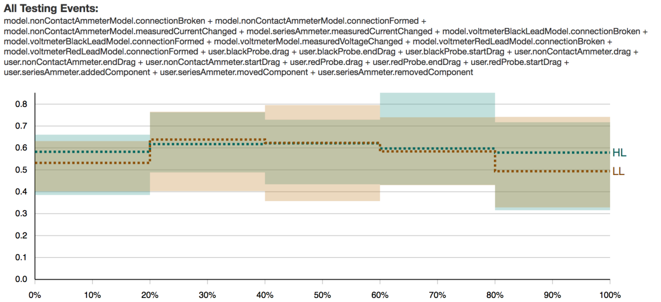


**Figure 5. Merging all testing results does not reveal interesting results. It appears as though HL and LL test similarly for the majority of interaction.**

reveal what critical testing events look like. In other words, what testing events correspond with productive learning, and how are they performed?

The heatmap visualization provides an overview of the differences between the two learner groups. Figure 3 shows the heatmap sorted by the last column, to emphasize differences between common actions of HL vs. LL during the last 20% of the interaction.

In regards to the types of testing, since the heatmap is sorted, we see that the top seven events are ones that HL perform more frequently than LL. 5 of those 7 relate to testing with the ammeter. Conversely, 7 of the 16 events that LL perform more frequently relate to voltmeter testing. Already, we see that there might be a difference in which testing instruments each group of students prefers.

In regards to the way testing was done, HL are both performing a larger amount of moving the ammeter device and measuring current changes with it. LL are the opposite: performing a larger amount of moving the voltmeter device and measuring voltage changes with it. Thus, the heatmap reveals different preferences with regard to the instrument, but not for the manner in which it is used. Next we examine these two questions in detail using the main Visualization and Merging panels, starting with the open question about the manner in which students test.

### *How does productive testing look like?*

A first step in trying to understand what testing events are important would be to assess whether active moving of the testing device is similar to observing a change in reading of the testing device while connected.

Here we describe testing with the ammeter as an example to answer this question. We can visualize this with Tempr without needing to do any merging as moving the testing device (user.nonContactAmmeter.drag) and the resulting change in reading of the testing device (model.nonContactAmmeterModel.measuredCurrentChanged) were logged actions in the CCK Simulation. It is important to note that drag events that end in connections being formed or broken lead to a change in reading. Thus, the two events are related. Figure 6 shows these two actions with Tempr. The lower graph, describing dragging, shows that HL tend to be dragging the ammeter more than LL. This is highly visible as the blue HL shading (representing the 75th percentile) extends above the LL brown shading. On closer inspection, we also see that the medians differ for dragging the testing instrument. For example, in the first 20% of interaction, both HL and LL measured current change for about 6% of the actions that took place (top chart). Additionally, HL also dragged the ammeter for 6% of the actions, while LL dragged the ammeter for only 4% of actions in that time period. This is a lower number than the number of actions that they observed current change, meaning that they observed more current changes than they actually moved the testing instrument. This is an indication of passive testing. It's likely that they left the testing instrument on the circuit as they were modifying the circuit, hence obtaining current changes without moving the device. One problem with this kind testing is that LL were not necessarily actually observing the changes in the testing device.

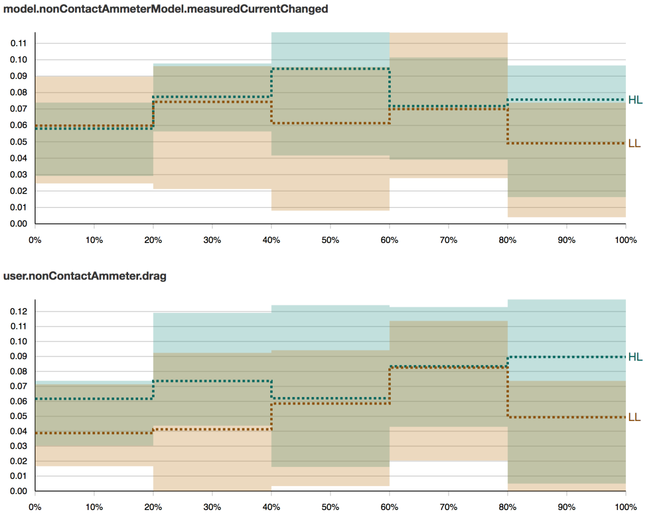


**Figure 6. Measuring current change with an ammeter vs actively using the ammeter. Low Learners drag the ammeter less frequently than they observe current changes, likely engaging in less explicit testing compared with High Learners.**

With Tempr, we were able to visually explore these two initially seemingly equivalent manifestations of testing and understand that, in fact, they are not the same. It also gave deeper insight into how the two groups test, allowing us to see that HL are engaging in more explicit testing while LL are partaking in more passive testing.

### *What testing instruments do different groups use?*

Based on the heatmap shown above, one can conjecture that HL students use the ammeter more than the voltmeter, while LL do the opposite. However, this conjecture is based on data from the last 20% of the interaction, across all testing events. Here we would like to evaluate this, focusing on the more significant active testing events. Our initial hypothesis was that HL test more than LL. To evaluate this, we merge all active movement of the three types of testing devices and plot this combination. Because we previously saw that moving the testing device is more important than detecting the measure-

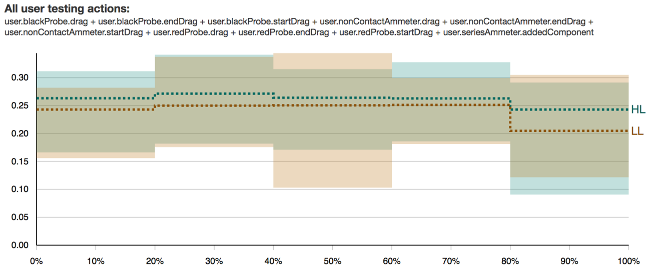


**Figure 7. Merging all testing device manipulation events. Note that HL and LL are nearly indistinguishable, with LL testing slightly less frequently than HL.**

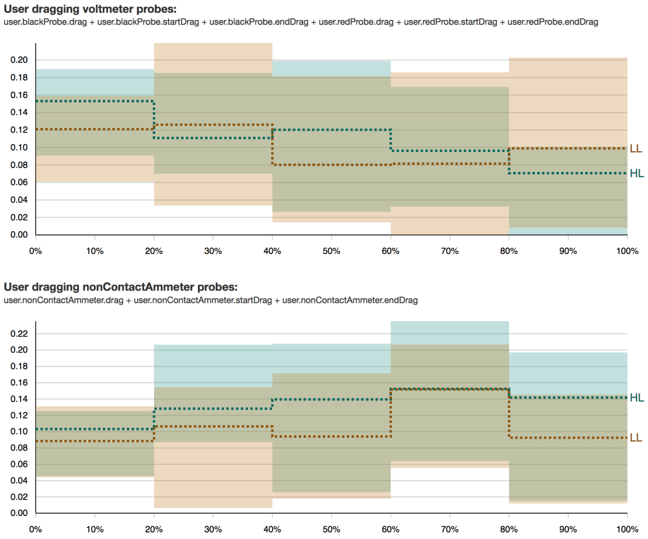


**Figure 8. The result of decomposing the different types of testing. Interesting to note that HL decrease their use of voltmeter (top graph) and increase their use of ammeter (bottom graph) over the course of the interaction, showing their ability to adapt to a new environment better than LL.**

ment change, we'll focus on these moving the testing device actions.

The result of this is shown in Figure 7. As can be seen, HL and LL are nearly indistinguishable. LL testing drops off at the end of interaction, but during the rest, the median lines are roughly equivalent. Both have quartiles that are roughly equivalent, with LL behaving less consistently in the middle of the interaction.

Since we did not find much difference in the previously explored level of abstraction, we can dive into each of the raw log events that comprise that merged event. Tempr allows us to easily try other combinations of features, such as manipulating the different types of testing events in the PhET CCK virtual lab: voltmeter and ammeter. If we visualize each of these with Tempr, the result is the graphs in Figure 8.

It is interesting to compare the difference in use of the ammeter and the voltmeter as it does appear that there are differences in how HL and LL are testing their circuits over time. We see is that the HL students increase the use of the ammeter over the course of interaction (see bottom graph in Figure 8).

The intent of Tempr is to reveal patterns that can be explained and understood. Revisiting the activity suggests a clear explanation for this finding. Since this activity is focused on resistors, the more useful testing instrument to use would be the ammeter. This is because changes to the resistor's resistance change the current but not the voltage. Additionally, we see that the HL use of the voltmeter actually decreased over the course of interaction (see top graph in Figure 8). That is, HL began using both instruments, and then favored the ammeter over the voltmeter. We can compare this to the LL whose use of the voltmeter only decreases very slightly from beginning to end of interaction. From this, it appears that HL are adapting to their environment. As mentioned, the students completed another activity before the one we are using the data from. This earlier activity was on understanding light bulbs, and the more fit testing instrument to use in that activity would have been the voltmeter. It makes sense that learners would start with using the voltmeter, since that was beneficial to their understanding in the first activity, but then that only the productive students would come to realize while interacting with the learning environment that the voltmeter would not be as useful here and that the ammeter was the better tool. LL instead continue to use the voltmeter, and appear to have not adapted as well to this activity.

## DISCUSSION AND CONCLUSION

We introduced a visualization tool that facilitates a novel exploratory analytic approach. It allows for the combination of bottom-up and top-down processes when engineering features to highlight differences in how different groups of students learn over time. We demonstrated, through a case study with the PhET CCK learning environment, how the tool help us (i) hypothesize what actions correspond with productive engagement, (ii) evaluate different sets of actions, (iii) compare groups of learners on these sets, and (iv) do so in the context of a temporal analysis.

With the aid of Tempr, we were able to use the heatmap to raise hypotheses and pinpoint the questions we wanted to answer regarding the instruments that students used (ammeter vs voltmeter) and the way in which testing was done (actively moving the device vs passively leaving the testing device connected). After investigation of these two with Tempr, we found that HL perform more active testing compared to LL. By visualizing use over time we also found that HL adapt more successfully to characteristics of the activity. This was found using Temper's ability to visualize combinations of raw events at different levels of abstraction - finding minimal differences in how students test when all testing events were merged, but discovering richer differences once we reduced to the different types of testing events.

We are currently using Tempr to analyze data from other complex environments such as MOOCs. While technically data from a variety of environments can be entered into Tempr, its utility across types of learning environments needs to be evaluated. For example, what is the dependency of Tempr on granularity of data? We hypothesize that Tempr is beneficial especially with data with high resolution, as a main advantage of the tool is its ability to identify and group sets of actions. However, this hypothesis is yet to be tested. It is also of interest to evaluate Tempr with hierarchical data. While using search terms in the merging panel can support the grouping of hierarchical data, presently Tempr lacks a structural support for levels in data. Tempr also focuses solely on ordinal information and lacks timing data. This choice was made to simplify its data structure, as Tempr takes simple lists of events. However, this compromises the ability to analyze by duration of events. This is relevant especially when data includes events of widely varying duration, such as video watching (often minutes) and problem attempts (often seconds) in MOOCs.

We are currently working to extend Tempr's capabilities to further support researchers in their ability to use data to understand more about how groups of learners are learning. Specifically, Tempr will soon support identifying and evaluating sequences of actions. For example, rather than merely merging actions (action *A or* action *B*), it may be of interest to sequence actions (action *A followed by* action *B*). This will allow researchers to use grammatical structures and reveal more complex patterns.

Overall, Tempr is not intended to replace the expert. Instead, it is a powerful tool to be used by experts who seek to obtain a more detailed understanding of learning trajectories in the target environment.

## ACKNOWLEDGMENTS

We would like to thank Kathy Perkins and the rest of the PhET team for their invaluable collaboration and feedback.

## REFERENCES

1. 2016. PhET Interactive Virtual Labs for Science and Math. (2016). `https://phet.colorado.edu/`
2. Vincent Aleven, Ido Roll, Bruce M McLaren, and Kenneth R Koedinger. 2016. Help helps, but only so much: research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education* 26, 1 (2016), 205–223.
3. Ryan S.J.d. Baker and Jody Clarke-Midura. 2013. Predicting successful inquiry learning in a virtual performance assessment for science. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 203–214.
4. Ryan S.J.d. Baker, Albert T Corbett, Ido Roll, and Kenneth R Koedinger. 2008. Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction* 18, 3 (2008), 287–314.
5. Michael Bostock. 2015. Visualizations with D3. (2015).
6. Michael Brooks, Saleema Amershi, Bongshin Lee, Steven M Drucker, Ashish Kapoor, and Patrice Simard. 2015. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*. IEEE, 105–112.
7. Bodong Chen, Alyssa F. Wise, Simon Knight, and Britte Haugan Cheng. 2016. Putting Temporal Analytics into Practice: The 5th International Workshop on Temporality in Learning Data. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge (LAK '16)*. ACM, New York, NY, USA, 488–489.
8. Cristina Conati, Lauren Fratamico, Samad Kardan, and Ido Roll. 2015. Comparing representations for learner models in interactive simulations. In *International Conference on Artificial Intelligence in Education*. Springer, 74–83.
9. Diansheng Guo. 2003. Coordinating computational and visual approaches for interactive feature selection and multivariate clustering. *Information Visualization* 2, 4 (2003), 232–246.
10. Hogyeong Jeong, Amit Gupta, Rod Roscoe, John Wagster, Gautam Biswas, and Daniel Schwartz. 2008. Using hidden Markov models to characterize student behaviors in learning-by-teaching environments. In *International Conference on Intelligent Tutoring Systems*. Springer, 614–625.
11. Josua Krause, Adam Perer, and Enrico Bertini. 2014. INFUSE: interactive feature selection for predictive modeling of high dimensional data. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 1614–1623.
12. Heidi Lam, Daniel Russell, Diane Tang, and Tamara Munzner. 2007. Session viewer: Visual exploratory analysis of web session logs. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*. IEEE, 147–154.
13. Zhicheng Liu, Yang Wang, Mira Dontcheva, Matthew Hoffman, Seth Walker, and Alan Wilson. 2017. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 321–330.
14. Sana Malik, Ben Shneiderman, Fan Du, Catherine Plaisant, and Margret Bjarnadottir. 2016. High-Volume Hypothesis Testing: Systematic Exploration of Event Sequence Comparisons. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 6, 1 (2016), 9.
15. Jeffrey D Marx and Karen Cummings. 2007. Normalized change. *American Journal of Physics* 75, 1 (2007), 87–91.
16. Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. 2013. Temporal event sequence simplification. *IEEE transactions on visualization and computer graphics* 19, 12 (2013), 2227–2236.
17. T. Powell. 2004. *JavaScript: The Complete Reference* (2 ed.). McGraw-Hill, New York, NY, USA.

18. Ido Roll, Ryan S.J.d. Baker, Vincent Aleven, and Kenneth R Koedinger. 2014. On the benefits of seeking (and avoiding) help in online problem-solving environments. *Journal of the Learning Sciences* 23, 4 (2014), 537–560.

19. Ido Roll, Ryan S.J.d. Baker, Vincent Aleven, Bruce M McLaren, and Kenneth R Koedinger. 2005. Modeling students' metacognitive errors in two intelligent tutoring systems. In *International Conference on User Modeling*. Springer, 367–376.

20. Ido Roll, N Yee, and A Cervantes. 2014. Not a magic bullet: the effect of scaffolding on knowledge and attitudes in online simulations. In *International Conference of the Learning Sciences*.

21. Michael A Sao Pedro, Ryan S.J.d. Baker, Janice D Gobert, Orlando Montalvo, and Adam Nakama. 2013. Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction* 23, 1 (2013), 1–39.

22. Ben Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations *(VL '96)*. IEEE Computer Society, 336–.

23. Jing Yang, Wei Peng, Matthew O Ward, and Elke A Rundensteiner. 2003. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*. IEEE, 105–112.

24. Jian Zhao, Zhicheng Liu, Mira Dontcheva, Aaron Hertzmann, and Alan Wilson. 2015. MatrixWave: Visual comparison of event sequence data. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 259–268.