Behaviormetrika (2018) 45:527–563
https://doi.org/10.1007/s41237-018-0063-y

ORIGINAL PAPER

How to conceptualize, represent, and analyze log data from technology-based assessments? A generic framework and an application to questionnaire items

Ulf Kroehne[1] · Frank Goldhammer[2]

Received: 28 February 2018 / Accepted: 9 August 2018 / Published online: 22 August 2018

Abstract

Log data from educational assessments attract more and more attention and large-scale assessment programs have started providing log data as scientific use files. Such data generated as a by-product of computer-assisted data collection has been known as *paradata* in survey research. In this paper, we integrate log data from educational assessments into a taxonomy of paradata. To provide a generic framework for the analysis of log data, finite state machines are suggested. Beyond its computational value, the specific benefit of using finite state machines is achieved by separating platform-specific log events from the definition of indicators by states. Specifically, states represent filtered log data given a theoretical process model, and therefore, encode the information of log files selectively. The approach is empirically illustrated using log data of the context questionnaires of the Programme for International Student Assessment (PISA). We extracted item-level response time components from questionnaire items that were administered as item batteries with multiple questions on one screen and related them to the item responses. Finally, the taxonomy and the finite state machine approach are discussed with respect to the definition of complete log data, the verification of log data and the reproducibility of log data analyses.

Keywords Log file analysis · Computer-based testing · Test development · Paradata · Response times · Finite-state machines

Communicated by Ronny Scherer and Marie Wiberg.

✉ Ulf Kroehne
kroehne@dipf.de

[1] Centre for Technology Based Assessment (TBA), German Institute for International Educational Research (DIPF), Schloßstraße 29, 60486 Frankfurt am Main, Germany

[2] Centre for Technology Based Assessment (TBA), German Institute for International Educational Research (DIPF) and Centre for International Student Assessment (ZIB), Frankfurt am Main, Germany

1 Introduction

Educational large-scale assessments are in the middle of introducing computer-based assessment and new methods of data collection. With this change of test administration mode, the incoming *log data* attract more attention, for instance, for the investigation of time on task (e.g., Scherer et al. 2015), to improve validity and reliability of computer-based administered measures (Ramalingam and Adams 2018), or to compare response sequences (e.g., He and von Davier 2015). Log data are not entirely new, in particular for questionnaires, as these additional data can be understood as part of the concept of *paradata* developed in the field of survey research (see, e.g., Kreuter 2013). Response times and response latencies constitute an overlapping area of research using paradata from surveys (e.g., Heerwegh 2003), psychometric research using log data from educational and cognitive assessments (see, e.g., Schnipke and Scrams 2002), applying techniques of educational data mining (e.g., Ma et al. 2016).

In the following, we will first refer to taxonomies and typologies of paradata (Callegaro 2012; Olson and Parkhurst 2013; McClain et al. 2018) showing that paradata and log data can be grouped on a surface level into three generic categories of *access-related*, *response-related* and *process-related* paradata. The taxonomy is not intended to be an exhaustive literature review, but rather an overview of examples that constitute the essence for categories and subcategories of paradata. Moreover, we will show that for some indicators the classification of paradata in terms of a taxonomy is not sufficient. Specifically, if indicators require the combination of multiple paradata points (e.g., multiple timestamps; Zhang and Conrad 2013) or a sequence of multiple log events, the (atomistic) classification of log events falls short.

On this background, we introduce the concept of *states* representing parts of the theoretically defined response process that should be distinguished from log events. The definition of states depending on specific research questions and assumptions about the targeted response process allows to combine and integrate single log events in meaningful and flexible ways.

Furthermore, we apply a concept from computer-science (so-called *finite state machines*) to develop a framework for the analysis of log data from technology-based assessments. The proposed framework serves to fill the gap between meaningful states and (multiple) log events by providing a method for the interpretation of states that are identified by observable behavior from raw process data (Zoanetti 2010). For that purpose, an abstract layer conceptualized with finite state machines is inserted between log events and indicators. The approach is developed to tackle what Luecht and Clauser (2002, p 76) called the real challenge in complex computer-based tests, "how to filter, encode, smooth and raw score the data to retain as much information as necessary for subsequent use, that is, to be combined to produce the outcome results".

Subsequently, we apply the generic framework to an empirical example from the context assessment of the Programme for International Student Assessment (PISA). This application relating item responses to item-level response times illustrates how those response times can be extracted although item batteries with multiple questions (i.e., items) on one screen were administered. The finite state machine approach enabled us to disentangle the time between different responses into time components that can be interpreted as item-level response times.

The closing discussion critically reflects selected benefits and restrictions of the taxonomy of paradata and the finite-state machine approach, in particular, with respect to completeness of log data, the verification of log data, and the reproducibility of log data analyses.

2 Taxonomy of *paradata*

To elaborate the need for an additional theoretical layer for the analysis of log data, we start with presenting a brief review of the literature on paradata. In survey research, *paradata* is known as data generated as a by-product of computer-assisted data collection methods (Couper 1998). Paradata include log data such as keystrokes, clicks and timestamps (Olson and Parkhurst 2013), gathered routinely in surveys. The taxonomy resulting from the following review represents our attempt to structure different types of paradata according to their use in a small number of categories, which we describe with prototypical examples.

2.1 Types of data

Callegaro (2012) distinguished four data types: *substantive data* (1), the results of assessments with one final response for each test taker to all administered questions or (sub)tasks. They are typically organized in rectangular datasets, and missing values are generally coded with specific pre-defined values. On the contrary, *paradata* (2) contain any information that describes how the data were collected, originally described as process data that come “for free” (see Olson 2013). We focus on paradata which describe data collections that directly involve the unit of observation.[1] The data are not necessarily rectangular (Olson and Parkhurst 2013) in the sense that (a) particular values might change during the assessment (for instance, for each session) and that (b) the data are fitting better into an event structure, where each event provides a possibly nested data structure that is specific to a particular event type (see, e.g., Kaczmirek 2009). Accordingly, *metadata* (3) describe the format and structure of the variables and provide “data about data”, e.g., codebooks, that are, however, often only available for substantive data. Some paradata, such as the response rate for a survey, become metadata when aggregated (Kreuter 2013). Finally, *auxiliary data* (4) are described as a separate type of data, not collected directly in the survey or assessment itself, but linked, for instance, at an aggregate

[1] Note that this differs in the categories used by Olson (2013), who created groups of paradata regarding the sample unit (neighborhood, housing, and person), the call record information and the observations recorded by the interviewer while interacting with the sampled person.

level. For educational assessments auxiliary data can arise, at the school level, such as the technical equipment and internet connectivity of the schools' computer pool.

Callegaro (2012) differentiates paradata describing the device type and paradata describing the questionnaire navigation. The typology of McClain et al. (2018) describes four phases (prior survey, recruitment, access, and response). We combine both views by reorganizing paradata into three main categories of *access-related*, *response-related* and *process-related paradata*,[2] partitioned in additional sub-categories (presented in Table 1 with prototypical examples).

2.2 Access-related paradata

Each time a participant is exposed to an assessment (e.g., a commissioned interviewer tries to reach a target person) *access-related* paradata are generated. Access-related paradata are often at least partially under control of target persons. This is in particular true for the time chosen by the target person to participate in an interview, survey or assessment as well as for the setting and environment, in which questions or test items are answered. Access-related paradata originate at the level of *persons* and can vary within persons over time (i.e., the paradata emerge for each intended, completed or interrupted *session*). Access-related paradata can be classified into three sub-categories: 'contact', 'setting' and 'device' (see Fig. 1), as will be detailed in the following.

2.2.1 Contact

This subcategory 'contact' collects information about the communication and interaction history, with contact information as an essential part (field process data). It will emerge as soon as the target person is called (call record, see, e.g., Hanly et al. 2016), invited by e-mail, visited by an interviewer or contacted via letter or postcard. Access-related paradata will also be generated whenever a target person visits a website to inform or participate in a web-based survey. Access-related paradata contain the administration mode itself and are mode-specific (Olson and Parkhurst 2013), and paradata might be missing due to mode-specific unit non-response (e.g., Klausch et al. 2013). 'Login' and 'resume' information is particularly relevant for web-based assessments that can be interrupted deliberately or due to technical issues (Sinharay et al. 2014). The participation status of a target person (e.g., not started, partially completed, completed) can be derived, from 'login', 'logout' and 'resume' information.

[2] The aggregated information from previous assessments classified by McClain et al. (2018) as prior survey phase represents auxiliary data, which can be linked if available. Besides, we disentangle response-related and process-related paradata to express the proximity of paradata dealing directly with the response and paradata dealing with the navigation.

Table 1 Categories and subcategories of the taxonomy of paradata with prototypical examples

Category	Subcategory	Description with prototypical paradata	Level
Access	Contact	Information about contact attempt; participation status; login and resume of interviews, surveys or tests; contact history	Person
	Setting	Location, time and time zone, environment and setting of an assessment or survey	Session
	Device	Properties of the hardware (e.g., type) and software (e.g., browser, plugins)	
Response	Answer	Answer selections and answer changes (i.e., input data directly related to answers); reaction times at the item level	Task/question
	Input	Mouse and touch events; keystrokes and input data (not directly related to answers); scrolling and zooming	Entity/screen
Process	Micro	Order of answers, navigation within entities, prompts and messages at the task level; request of additional information; time measures at the entity level	Entity
	Macro	Last entity and drop-off, navigation between entities, item or question review	Test section

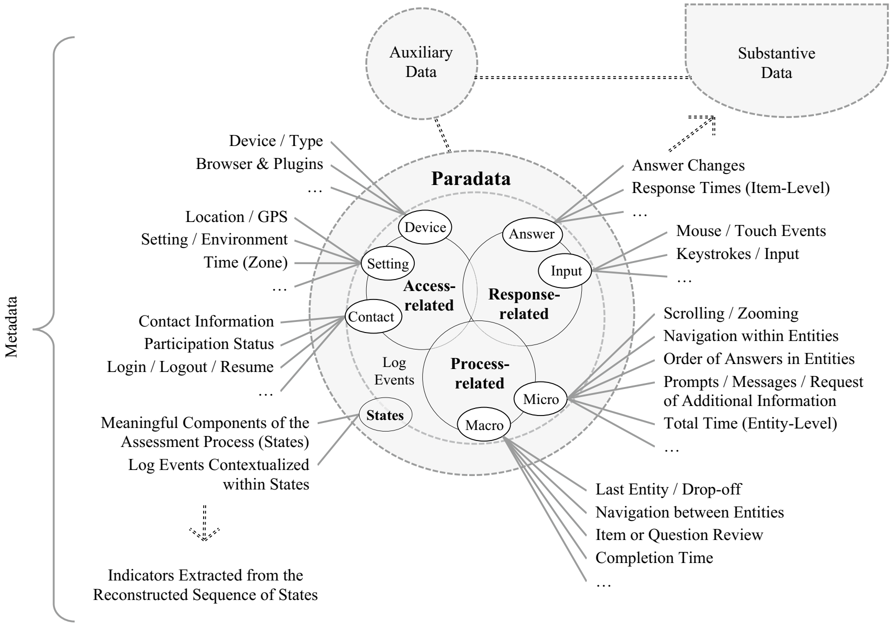


Fig. 1 Taxonomy of paradata with categories access-related, response-related and process-related paradata

2.2.2 Setting

All information describing the specific conditions of a data collection is subsumed in a subcategory with non-device specific characteristics of the environment of a data collection. 'Setting' includes location related information, for instance, gathered with mobile devices using the built-in GPS. Either self-selected or arranged with an interviewer, the participation time (e.g., Durrant et al. 2011) and the time zone of test takers for a particular access to an assessment belong to this subcategory.

2.2.3 Device

The subcategory 'device' represents technical information depending on the testing mode and the deployment technique (see, for instance, Dadey et al. 2018, for a review of effects across different devices). It includes hardware-related information about the device type (e.g., desktop, laptop or mobile; see Schroeders and Wilhelm 2010), and the screen (orientation, pixel width and height, density, size in inch; see, e.g., Bridgeman et al. 2003). Included in this category are also information about software components, such as the version and specific features of the web browser (e.g., plugins, permissions, installed certificates, etc.) as well as properties of the infrastructure such as the network connection (type, bandwidth, latency, etc.), if relevant for a web-based assessment (see, e.g., Bennett 2015).

2.3 Response-related paradata

Response-related paradata are defined as events gathered as by-product of collecting responses. Each event x (e.g., the *answer change event* that indicates that the selected answer for a particular question was changed) is defined by an event type out of the set of all event types Σ as well as by a set of properties (e) that are provided for an event of that type (e.g., the properties that for test taker i the answer to question j was changed to the value v).

2.3.1 Answer

The most obvious paradata of this subcategory are answer-change events to questions, tasks or items. For simple item formats (e.g., multiple choice questions such as QTI *choice interactions*, IMS 2012), log events contain the raw response to an item. For more complex item formats (e.g., items requiring multiple interactions such as the QTI *order interactions*), log events might contain only the incremental difference of the status before and after an answer-change, and the trace of all log events is required to reconstruct the final raw response.

Typically, answer-change events contain a time measure (either a timespan relative to the item start or a timestamp). If tasks or questions are administered in a way that allows a direct interpretation of the time measure, as it is, for instance, in the so-called "One Item, One Screen" design (OIOS; e.g., Reips 2002), the time differences between the event indicating the appearance of a question on screen and the answer-change events can be used as item-specific time measure (*response time*). Paradata of the subcategory 'answer' can, for instance, be data from a computer-assisted personal interviewing (e.g., Couper and Kreuter 2013) and strategies such as the "four-screens-per-question-technique" (e.g., Mayerl 2013) allow to measure *response latencies* in interviews. For item batteries, response times are either not analyzed at all (e.g., Yan and Tourangeau 2008), the *total time per page* is used (Malhotra 2008; Mavletova and Couper 2016; Höhne and Schlosser 2018), or the *completion time* for the whole instrument is analyzed (e.g., Couper et al. 2013; Liu and Cernat 2016). Only some exceptions investigate time differences between questions of item batteries (e.g., Zhang and Conrad 2013).

2.3.2 Input

Not all user interactions necessarily result in an answer change. The subcategory 'input' includes further log events indicating that a test taker interacts with the assessment platform although these events do not directly result in a changed response, but provide additional information about the test takers' behavior. Events indicating mouse move or touch move are collected typically in terms of coordinates captured with a concrete sampling rate (e.g., Stieger and Reips 2010). However, additional events associated with information displayed on screen, such as mouse over or hoover of, for instance, buttons or links, can also be classified into subcategory ‘input’ (e.g., Khasawneh et al. 2012).

2.4 Process-related paradata

Process-related paradata, include all pieces of information that arise in the course of an assessment going beyond answers and inputs. Without limiting the general scope, we consider computer-based instruments on two levels as tasks or questions that can be grouped into larger ‘*entities*’. Entities can be formed from screens, pages, items or tasks but not each element necessarily requires a *response*. In simple computer-based instruments, an entity can consist of only one single item (e.g., OIOS).

2.4.1 Micro

The subcategory ‘micro’ represents process-related paradata *within* entities. If the information presented on pages or screens is modified either by scrolling or by zooming, related log events are classified as ‘micro’ (see e.g., Higgins et al. 2005, for the effect of scrolling on reading test performance). If entities create, for instance, unit-structured test, the navigation between pages or screens contributes to process-related paradata of the subcategory ‘micro’. If questions are combined in entities as in item batteries, the order of answers within entities also belongs to ‘micro’. The order of answers within entities can be derived using multiple timestamps of answer-change events (e.g., Heerwegh 2003).

The appearance and disappearance of prompts or messages during the assessment is included in the subcategory ‘micro’. Logging of those is not only important to reconstruct all information visible on the screen. In particular, the so-called *modal dialogs* are important as they are hindering test takers from interacting with the instrument while open. This effectively reduces the available time for completing tasks. Similarly, logging the request for additional information (e.g., on-screen help for questions or tasks) is required to capture the whole test-taking process. Finally, as mentioned above, for item batteries time-related paradata are most often not considered at the item-level and measures such as the *time per page* fall into the subcategory ‘micro’.

2.4.2 Macro

Whereas subcategory ‘micro’ contains paradata *within* entities, the subcategory ‘macro’ addresses process-related paradata at the higher level. Often, there is a second level of navigation that can be described as navigation *between* entities (see, e.g., Luecht and Sireci 2011), and while navigation within entities is typically unrestrained, navigation between entities is often restricted and managed by the assessment platform. A specific type of navigation between entities is the *item* or *question review* (i.e., the possibility to revisit already visited entities). If review is permitted in a particular assessment, process-related paradata in the subcategory ‘macro’ allow to retrace test takers’ usage of the offered opportunity. Item review can be deduced from the order of answers (subcategory ‘answer’ of response-related paradata), only if answers are changed in the revisited tasks.

For tests administered without proctors (e.g., unstandardized online assessment; Kroehne et al. 2018), the last entity before interrupting, canceling or aborting an assessment (drop-off) is part of the subcategory ‘macro’, while the related login for continuing an interrupted session might also be considered as access-related paradata (‘contact’). Further process-related paradata within (‘micro’) or between (‘macro’) tasks and questions are defined by the testing interfaces, e.g., the availability and usability of additional computer-based tools (Way et al. 2015). Time measures at the level of the instrument (e.g., the *completion time*), are considered as paradata of this subcategory as well.

3 States

The taxonomy described so far includes paradata related to access, response and process that can be illustrated with prototypical examples. All of them share the notion that information encoded in the log data can be identified primarily from the *type* of the log events. Introducing the concept of *states* goes beyond this direct relationship between the occurrence of events and the meaning of the information, by contextualizing log events in meaningful components, labeled as states. Potential usages of log data and advanced applications of log data analyses can be understood as the decomposition of a response process into such states, which provide the theoretical foundation for the definition of indicators. The decomposition of the test taking or survey process into states and the reconstruction of the sequence of states using log events provide an abstract theoretical layer between the platform-specific log events and the interpretation of derived indicators.

By adding ‘states’ as a distinct category of the taxonomy (see Fig. 1) we emphasize the importance of meaningful sections of the response and test-taking process. These states create contexts, in which log events can be interpreted and accordingly extend the taxonomy to indicators that go beyond the use of information that is encoded in the event type. Theoretical considerations about the interplay between respondents and the assessment platform like the expectation, that the question stem and the response categories are read before the first response to an item battery is given, constitute the meaning of states.

Ambiguity in the distinction of access-, response- and process-related paradata, illustrated in Fig. 1 as overlap between the categories, highlights the need to concretize the intended interpretation and use of log data for a particular assessment. As explained by Kaczmirek (2009), at the lowest level (labeled by Kaczmirek as first level paradata), records of single events are, first of all, technical in nature. One approach of grouping paradata to concepts (second level) and aggregating across ‘variables *or* persons’ (third level) or across ‘variables *and* persons’ was suggested by Kaczmirek (2009). We pursue an alternative approach to achieve an in-depth view regarding the relationship between log events and conceptually relevant features of the response process by the identification of meaningful components of the process (i.e., states).

Which states are relevant and how the desired indicators can be meaningfully extracted from the sequence of states must be defined in an *assessment framework* (e.g., Goldhammer and Zehner 2017; Mislevy et al. 2012; OECD 2016), going beyond simple taxonomy of paradata with prototypical examples. Such a framework elaborates the targeted (latent) construct (e.g., cognitive states of information processing), identifies observable evidence for that construct and the item content needed to elicit the desired behavior captured by considered states. This requires integrating substantive domain knowledge into a study-specific description of the distinguished states and the planned interpretation and use of log data and derived indicators.

The need for an additional layer can also be illustrated by the computation of item-level response time measures for item batteries. As mentioned, existing research using such response times computed them using ad hoc definitions, for instance, "as the elapsed time between submission of an answer to the previous and current question, based on timestamps collected on the server" (Zhang and Conrad 2013, p 128). Hence, information from the paradata stored as log events is extracted (Heerwegh 2003). However, considering only answer changes between subsequent responses ignores parts of the empirical phenomenon. For instance, the time prior to the first response to an item battery is expected to contain also the time to read the question stem and the response categories. If different states of the response process are not differentiated, the meaning of the resulting item-level response time measure is different, for instance, for cases omitting the first item and for cases answering the first item. However, by distinguishing different states for reading the stem and answering the first question of an item battery versus answering subsequent questions of the same item battery, the approach we present in the next section can overcome this imprecision of ad hoc definitions.

4 Using finite state machines to analyze log data

In the following, we describe a formal approach as a theoretical layer intended to bridge the gap between (arbitrary) log events that are specific for a technical implementation of an assessment platform and derived indicators related to substantive research questions of log data analysis. For that purpose, we use the well-known idea of *finite-state machines* or finite state automata, which represent a formal mathematical model of computation, and apply the concept of abstract machines to the interaction between test taker and assessment platform. The additional theoretical layer is related to the taxonomy presented in the previous part, as it allows to define states based on log events and to aggregate information about states to indicators (see Fig. 2).

4.1 Decomposing processes using finite-state machines

Finite-state machines (FSMs, e.g., Alagar and Periyasamy 2011) are already used in assessments to program and create complex, interactive instruments

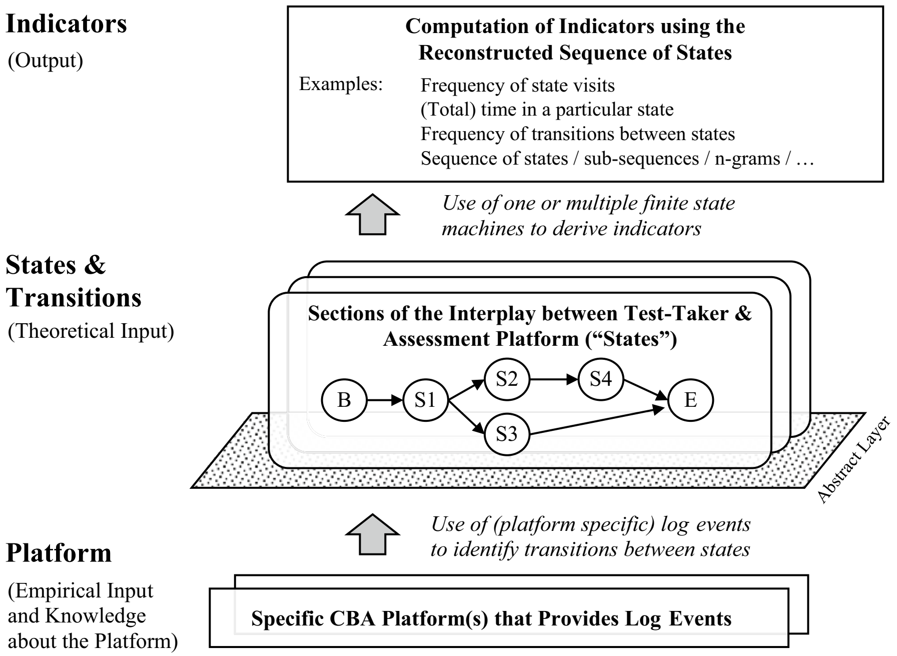

Indicators
(Output)
Computation of Indicators using the Reconstructed Sequence of States
Examples:
Frequency of state visits
(Total) time in a particular state
Frequency of transitions between states
Sequence of states / sub-sequences / n-grams / …
Use of one or multiple finite state machines to derive indicators
States & Transitions
(Theoretical Input)
Sections of the Interplay between Test-Taker & Assessment Platform ("States")
B
S1
S2
S3
S4
E
Abstract Layer
Use of (platform specific) log events to identify transitions between states
Platform
(Empirical Input and Knowledge about the Platform)
Specific CBA Platform(s) that Provides Log Events

Fig. 2 Illustration of states between log events and indicators computed using reconstructed sequences of states

(see, e.g., Roelke 2012; Neubert et al. 2015), and are well known as a technique for software development, used, for instance, in game-based assessments (Mislevy et al. 2016). In the following, we use FSMs as a tool to analyze log data retrospectively, that is, after the assessment is finished. Note that using FSMs as an analysis tool means that neither the assessment platform (i.e., the software used to administer the computer-based test or questionnaire) strained with additional load nor is it necessary to know the specific FSM before or during the data collection. Analyses using FSMs have been rarely applied to log data, for instance, Almond et al. (2012) used FSMs to classify log entries, but focused on processing of keystrokes in a writing assessment. Bergner et al. (2014) analyzed data from a complex computer-based task of engineering literacy assessment using state sequences as modeled with the R package TraMineR (Gabadinho et al. 2011) without explicitly linking the investigated states to the log events.

The proposed framework is a new approach to contextualize the information provided as log events in states. The framework generalizes the retrospective analysis of Almond et al. (2012) beyond the classification of typing events and elaborates the retrospective reconstruction of state sequences as a versatile and generic tool for the analysis of log data using different FSMs to extract specific indicators depending on the respective research question.

4.1.1 States and set of states

As described, states designate specific sections of the process conceptualized as an interplay between test taker and assessment platform and require a formal definition (Almond et al. 2012). The meaning of states is constituted by three components:

a) the information that is presented by the assessment platform in a designated phase of the assessment (e.g., the texts, images, videos etc. shown on the screen),
b) the possibilities to interact with the content offered by the platform in a that specific phase (e.g., the interactive components such as buttons and input elements), and
c) a justifiable theoretical interpretation of the meaning of this particular period of an assessment (e.g., expected cognitive processes that are relevant for the state with respect to the interplay between test taker and assessment platform).

In this sense, states provide the semantics for a theory-based analysis of log data. Describing and defining states that are distinguished for creating indicators is seen as the cornerstone in log data analysis. Likewise, for the computation of simple descriptive statistics or the application of complex psychometric methods, such as process mining (e.g., Romero 2011; Ferreira 2017), a terminology is needed that relates gathered log events to meaningful parts of the assessment process. States are not more than *labeled eggs*—similar to constructs in latent variable models (Nachtigall et al. 2003)—until a proper description is provided and evidence is gathered that support the hypothesized meaning of the processes related to this state. It is important to emphasize that the meaning of states is not created *bottom up* from information and provided interactivity of the assessment platform (a and b), but rather from the *top down* description of the states (c) related to the assessment framework (see previous section). Accordingly, the concept of states in this framework goes beyond the use of FSMs as an algorithmic tool for implementing or modeling complex systems.

States are thought as the *filtered* data that *encode* information that is used for the computation of indicators (with respect to Luecht and Clauser 2002, see above) by contextualizing events. Hence, the same log event can be understood differently in the FSM approach, depending on the current state.

No empirical data are required for *defining* the set of states S that are used to compute indicators for empirical applications. States can be defined a priori, and this even should be done based on an assessment framework to make sure that the assessment system finally provides all log events needed for identifying those states. For the a priori definition of states, no knowledge about the assessment platform is needed, so that the definition of states is not expected to be specific for a chosen implementation for a computer-based item (top down). However, states can also be defined or changed afterwards, for instance, to analyze existing log data (as we will show in the empirical application). The framework described in this paper can be used for any log data, as the reconstruction of the sequence of states is performed as a first step in the analyses of log data. From this follows that indicators computed from states are expected to be comparable between different assessments when defined with respect to identical states, while indicators directly computed from the log events are prone to contain platform-specific characteristics.

To reconstruct the sequence of states using available log data from a completed assessment, stored events are required to differentiate between states. Accordingly, not each set of states S can be analyzed using platform-specific log events Σ from every platform. However, as said, the definition of states is intended to be independent from the so-called *input alphabet* $\sum$ (i.e., all log events provided by an assessment platform). The considered states should be motivated and described theoretically and not narrowed to the specific characteristics of log events available from an assessment platform.

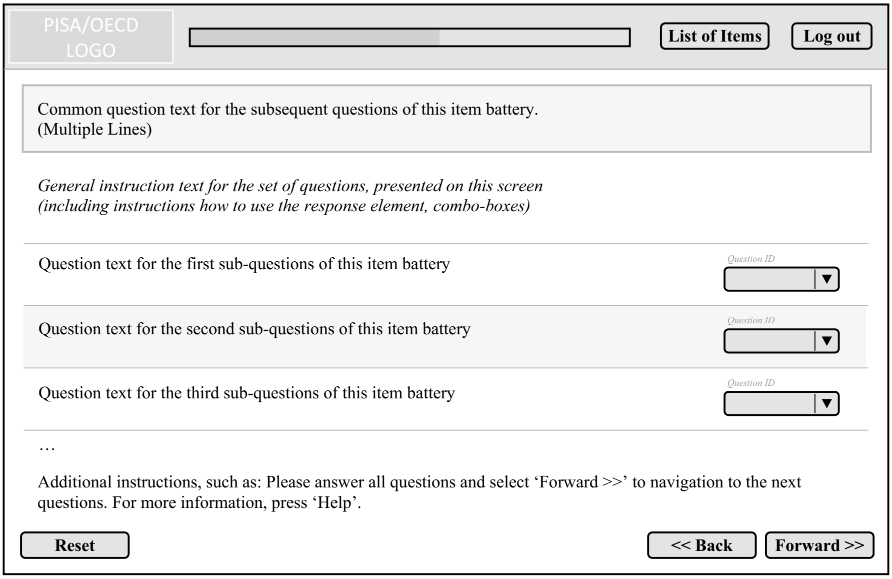
4.1.2 Definition of a FSM

A FSM $M\langle S, s_0, \sum, \delta, F\rangle$ is defined by a finite number of states (i.e., a finite, non-empty set of states S), deterministic transitions between states (i.e., we use deterministic state machines) and triggers that provoke a particular transition from one state to another. FSMs starts with an initial state (s_0) and are only in one state at a time (*current state*). The set Σ is the input alphabet, and δ represents the *state-transition function* (i.e., a definition of the possible transitions between states). The FSM is expected to end in an *accept state* out of a set of final states F (a subset of S), when the stream of all input elements (i.e., the list of log events of type $x \in \Sigma$ for a test taker) has been processed successfully, event by event.

4.1.3 Transitions between states

Transitions are either triggered by internal events (such as timers) or external events (such as button clicks). Log events processed by the FSM as the input alphabet $\sum$, can be used to *identify* transitions and thereby states. Transitions are represented in the formal description of a state machine by a state-transition function δ. This function is typically called *partial* state-transition function $\delta(q, x) \to q'$, because it only defines state transitions between states q, q' and *selected* triggers $x \in \sum$. The state-transition function returns, for a current state $q \in S$ of M, the new state $q' \in S$, when trigger $x \in \Sigma$ occurs. Consequently, the transition triggered by a log event $x \in \Sigma$ depends on the current state q. Especially, this property makes state machines a valuable tool for log data analysis, as it contextualizes the meaning of log events x (e.g., pressing the back button of a web browser) with respect to the FSMs' current state $q \in S$ (e.g., the current page).

FSMs can be visualized by directed graphs, typically called *state diagrams*. States are represented by circles and transitions are represented by arrows. For the analysis of log data, the arrows are linked to the triggers (e.g., log events) that are used to identify the transitions.



Common question text for the subsequent questions of this item battery.
(Multiple Lines)

General instruction text for the set of questions, presented on this screen (including instructions how to use the response element, combo-boxes)

Question text for the first sub-questions of this item battery

Question text for the second sub-questions of this item battery

Question text for the third sub-questions of this item battery

…

Additional instructions, such as: Please answer all questions and select ‘Forward >>’ to navigation to the next questions. For more information, press ‘Help’.

Fig. 3 Example CBA screen of the PISA 2015 context questionnaire (schematic view) for an item battery with 3 items

4.1.4 Extensions (guards, variables and look-ahead)

For practical applications of FMS’s, the trigger used in the state-transition function does not necessarily only refer to a specific event type $x \in \Sigma$ but also to additional information specific for particular event types. Such properties of events denoted as (e), for instance, the specific question an answer-change event belongs to, can be used to formulate conditions (*guards*) that must be fulfilled, that M accepts an event $x \in \Sigma$ in state $q \in S$. Moreover, extensions with respect to *variables* (known as extended state machines) can be used to identify state transitions with sparse log data. Finally, when log data are analyzed retrospectively with FSMs, guards that inspect not only the current log event, but incorporate all (or all remaining) events for an individual test taker (*look-ahead*) can be used practically (see the Table 3, below, for examples, i.e., is_last_event, nearest_event_is).[3]

4.2 Computing indicators using state machines

Using FSMs allows defining indicators with respect to the set of states S by combining theoretical input with empirical input [log events $x \in \Sigma$ with additional event-data

[3] Using look-ahead technically requires an extended definition of finite state machines. Instead of adding this additional complexity we formulate specific *guard operators* that evaluate to *true,* if M in a given state $q \in S$ accepts $x \in \Sigma$, using $\delta(q, x)$, the list of tuples $\langle i, t, x, (e) \rangle$ containing all log events for test taker i, and a variable j that points to the current element in the list of tuples.

(e) and timestamps t] and knowledge about the platform and the implementation of computer-based tasks (see Fig. 3). This can be conducted for the test takers separately, each time starting with the state s_0, and it is expected that the FSMs for each test taker reach one of the end states $f \in F$.

4.2.1 Reconstructed sequence of states

Using FMSs allows reconstructing how a test taker followed through the *sequence of states* $q \in S$, distinguished in a particular FSM. Based on the identified states various indicators can be computed. To include time into the FSM approach, timestamps that are provided with the log events can be used.

Thus, the FSM approach disentangles processing log events (this is done using the FSM) and the computation of indicators (this is done using the reconstructed sequences of states). For empirical log data analysis, this offers the possibility to include paradata comprising multiple events in a coherent way. More specifically, it fosters the separation of steps required to parse and read the log data (i.e., the empirical input) from the steps used to extract meaningful indicators (Heerwegh 2003).

4.2.2 Augmented log data

The list of tuples $\langle i, t, x, (e)\rangle$, that represent the empirical input for test taker i, is augmented with additional information from the reconstructed sequences using a FSM M as follows: (1) the state $q \in S$ of the FSM before an input element $x \in \Sigma$ was processed (starting with s_0 for the first tuple), (2) the current state $q' \in S$ of the machine after a $x \in \Sigma$ was processed and (3) the relative time difference *td* to the previous log event (starting with zero for the first tuple in the list).[4] Each tuple $\langle x, t, i, (e)\rangle$ in this list represents a log event of type $x \in \sum$ from the input alphabet that occurred at time t and belongs to a test taker $i = 1 \ldots I$. To reconstruct the sequence of states, the list of tuples is processed event by event and augmented with q, q', td,[5] starting with s_0 for each test taker (see Table 3, below, for an example).

In general, indicators derived from log data using FSMs can be formulated as different aggregates of the reconstructed sequences of states in the augmented log data. Discussing and elaborating all possible ways to compute indicators is beyond the scope of this paper. Instead, in the following, we describe three outputs of the FSM approach that provide the source for different types of indicators: the *sequence of states*, the *state summary table*, and the *state transition table*.

[4] The time difference to the previous log event td is added for convenience to each tuple in the list to simplify computation of the total time on states as the sum of td for all tuples with a particular state q'.

[5] Note, neither the timestamp t nor the placeholder term (e) conflicts with the narrow definition of M as long as no information from t or (e) is used in $\delta(q, x)$ to reconstruct the sequence of states. However, as soon as *guard operators* are included, an extended definition of finite state machines would be required formally.

4.2.3 Sequence of states and sub-sequences

Concatenating all states of a test taker allows to extract the sequence of states, which can represent an indicator itself when states are focused that can occur in different meaningful orders, for instance, to identify problem-solving strategies (e.g., Tóth et al. 2014). The sequence of states can also be used to cluster test takers with respect to sequences (e.g., using edit distance as in Hao et al. 2015). Beyond the complete sequence, subsequences of a specific length can be counted automatically as output of the FSM approach, for instance, ordered as n-grams (e.g., He and von Davier 2016).

4.2.4 State summary table

A state summary table can be created for each test taker from the augmented log data, containing all defined states, the frequency how often states were visited, the total time on each state and additional measures for each state, such as the time of the shortest and longest visit. Values of binary indicators or count indicators, such as indicators for the *relevant page visit*, the request of *source information* and for *tool use* can be directly metered from the state summary table. For states that occurred at least once, the *time on state* can be used to compute values for further metric indicators, such as the *time on task* (Goldhammer et al. 2014), the *reading time* (e.g., Richter and Naumann 2000) and the *edit time* (Almond et al. 2012).

4.2.5 State transition table

Summarizing the augmented log data for each test taker with respect to rows that contain different values in the state before (q) and the state after ($q^{'}$) allows to create a state transition table, that counts the frequency of the directed transitions from one state to another. From the state transition table indicators that refer to the transition between states, e.g., the frequency of backward navigation from questions to the stimulus, can be extracted. Moreover, the state transition table can be used to create an aggregated representation of the navigation between states, such as an adjacency matrix.

5 Empirical application: item-level response times from item batteries

In the following empirical example, we first demonstrate how the FSM approach can be used to extract item-level time components from the PISA context questionnaire assessing ICT familiarity (Jude 2016). Then, we relate the extracted item response times to the item responses to explore whether there is any systematic (non-)linear relationship between the time needed for answering a question and the given answer. Thus, this empirical analysis serves to illustrate the value of the framework for conceptualizing, representing and analyzing log data.

Item-level response times from questionnaire items (e.g., Likert type) are an interesting source of information above and beyond item responses. For instance, item response times can be used to evaluate the data quality (e.g., Wood et al. 2017). Furthermore, in personality assessment, response times have been decomposed into various pieces to address how the fit of trait level and item difficulty affects response behavior (e.g., Ferrando and Lorenzo-Seva 2007; Molenaar, et al. 2015). The distance-difficulty hypothesis (e.g., Eisenberg and Wesman 1941) predicts that response time—as an indicator of the difficulty of responding to an item—increases with decreasing distance between person (i.e., trait level) and item (i.e., difficulty).

Item-level response times can be easily obtained when the OIOS-design (Reips 2010) is used. However, the PISA 2015 context questionnaire was administered with item batteries combining multiple items of a scale on *one* computer-screen (see Fig. 3 for a schematic visualization).

Note, that the questionnaire screen shown in Fig. 3 offered also different possibilities to navigate (buttons 'List of Items,' 'Back' and 'Forward'), contained a shortcut to reset all answers (button 'Reset') and it was possible to log-out from the assessment.

5.1 FSM representing state sequences in item batteries

The log events provided by the TAO platform (see OECD 2017, for information about the technical implementation of the assessment) represent the input alphabet for the FSM approach.

The log events can be classified with respect to the taxonomy of paradata (see Table 2). Furthermore, the events in Table 2 were considered to identify transitions of a FSM for the item battery (i.e., for all items of a particular scale administered on one computer screen) as shown in Fig. 4. This FSM differentiates between the substantive states 'Reading (stem) & first questions', 'Remaining questions' and 'Time after last answer'. Unfortunately, the available log data are incomplete regarding this FSM (because no log event indicates, when a requested list of items is closed). Accordingly, the transitions between 'List of items' and 'Reading (stem) & first question' and 'Remaining questions' are not identified and a FSM that explicitly separates the state 'List of items' from other states cannot be modeled.[6]

Time between the events 'ITEM_START' and the first answer-selection event can be interpreted as time for 'Reading (stem) & answering first question'. The time after the first response is interpreted as time for answering the 'Remaining questions' (i.e., as a time measure that should correspond to the sum of all item-level response times of the remaining items). Finally, the state 'Time after the last answer' is modeled separately. This time component is not related to single responses rather to an

[6] Note, that 26 cases have been removed from the analyses which requested the 'List of items' in the investigated item battery.

Table 2 Description of log events provided by the assessment platform available for the extraction of item-level response times

Subcategory		Raw log event	Description
Response related	Answer	ANSWER_SELECTION	Event indicating that an answer was changed including timestamp t; including an identifier of the question as additional event-specific information (e)
		RESET	Event indicating that all responses to the current screen were removed
Access related	Contact	SELECTED_LOG_OUT/LOG_OUT	Events indicating that a log-out was requested/performed
		SESSION_START	Event indicating that a new session was started (i.e., a previously interrupted session was continued)
Process related	Micro	ITEM_START	Event indicating that the item was started
	Macro	SELECTED_FORWARD/MOVE_FORWARD	Event indicating that the test taker clicked the button “Forward”/that the screen was exited to the next screen
		SELECTED_BACK/MOVE_BACK	Event indicating that the test taker clicked the button “Back”/that the screen was exited to the previous screen
		LIST_OF_ITEMS/SELECTED_JUMP/JUMP_ITE	Event indicating that the test taker clicked the button “List of Items”/that the screen was exited to a screen selected from the list of items

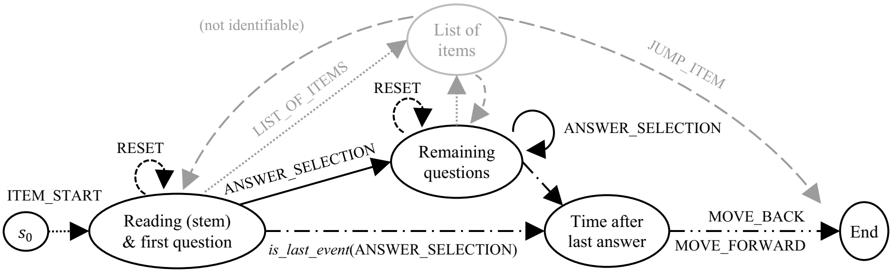


Fig. 4 State diagram for a FSM differentiating states “Reading (stem) & first question”, answering “Remaining questions” and

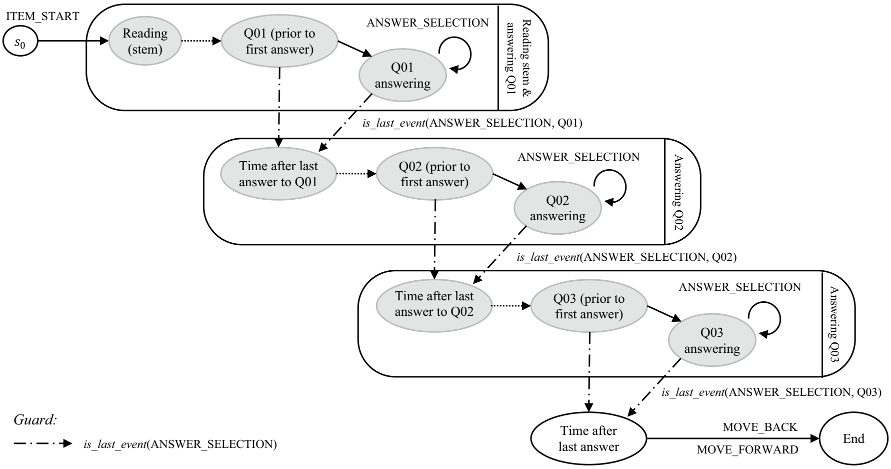

ITEM_START
s_0
Reading (stem)
Q01 (prior to first answer)
ANSWER_SELECTION
Q01 answering
Reading stem & answering Q01
is_last_event(ANSWER_SELECTION, Q01)
Time after last answer to Q01
Q02 (prior to first answer)
ANSWER_SELECTION
Q02 answering
Answering Q02
is_last_event(ANSWER_SELECTION, Q02)
Time after last answer to Q02
Q03 (prior to first answer)
ANSWER_SELECTION
Q03 answering
Answering Q03
is_last_event(ANSWER_SELECTION, Q03)
Guard:
is_last_event(ANSWER_SELECTION)
Time after last answer
MOVE_BACK
MOVE_FORWARD
End

Fig. 5 Schematic view of the construction of the states “Reading stem & answering Q01”, “Answering Q02”, “Answering Q03” and “Time after last answer”

overall speed level for processing the questionnaire and the certainty of answering all questions of a screen.

To decompose the state for answering the ‘Remaining questions’ an extended FSM can be developed. The implementation of this FSM can be simplified using a guard ‘*is_last_event*’ that evaluates to true for a current event, if subsequent to a considered event no further ‘ANSWER_SELECTION’-events are present in the list of log events. Figure 5 shows the idea to split the answering process by ‘ANSWER_SELECTION’-events to extract item-level response times: The picture is not a complete state diagram, but illustrates the generalization of the FSM shown in Fig. 4 for an item battery with only three items. Although the available log data do not allow differentiating between ‘Reading (stem)’ and the two states for working on question one [i.e., the states ‘Q01 (prior to first answer)’ and the optional state ‘Q01 (answering)’ when the test taker changes the answer to question one], we can identify the combined state ‘Reading stem & answering Q01’ for all test takers, that start

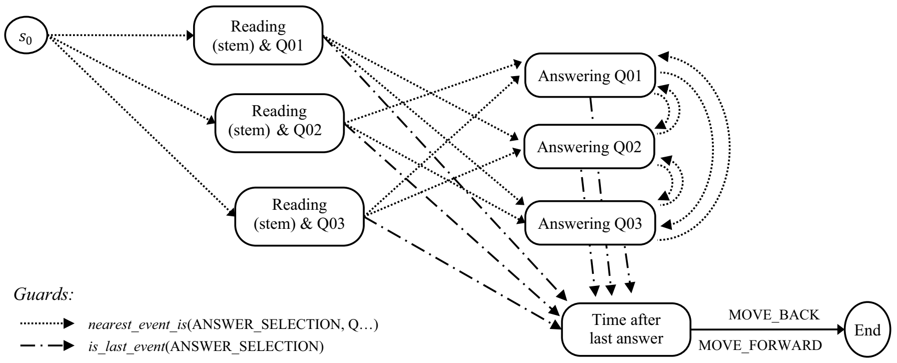


Fig. 6 State diagram for a FSM differentiating states "Reading & Q…", "Answering Q…" and "Time after last answer" used for the decomposition of item-level response times

working on the item battery with question 'Q01', and that answer the three questions in the simple sequence 'Q01–Q02–Q03'. For these test takers, we can also identify a state 'Answering Q02' that includes the 'Time after last answer' to the previous question by identifying the transition with the guard '*is_last_event(ANSWER_SELECTION, Q2)*' that evaluates to true, if the next answer-selection event is not related to question 'Q02'. In the same way, time measures for all remaining questions can be identified.

To apply this approach to all test takers (i.e., to all potential sequences), we formulate a FSM that contains two states for each question: on state for the question answered as the very first question on a screen (including the time prior to the first response that contains the time for reading the stem, 'Stem & Q1'…'Stem & Q3') and a second state for each item answered after the stem was read and the first question was answered. The state transition function includes only transitions from the states 'Stem & Q1'…'Stem & Q3' to the states 'Q1'…'Q3'. To simplify the implementation of the state machine, we defined an additional guard operator '*nearest_event_is*', which was used to switch to the appropriate state according to the question ('Q1', … 'Q3') to which the next ANSWER_SELECT-event belongs (see Fig. 6 and Table 3, below).

5.2 Method

5.2.1 Data

We use data from the PISA 2015 context assessment from Switzerland for one item battery of the ICT familiarity questionnaire (IC008: 'How often do you use digital devices for the following activities *outside of school*?', see Appendix) that was administered with 12 questions on one screen. The 91.189 log events (see Table 2) generated by 5.736 students were used as input to an FSM. For the analysis, we created a list of tuples $\langle x, t, i, (e) \rangle$ described above from XML files provided by the platform for each test taker for the selected battery.

Table 3 Annotated example table with raw log data for one test taker starting with question "Q02" and augmented information from the FSM approach

Line	Raw log data					Augmented information from the FSM Approach		
	Person-identifier	Timestamp	Event name	Question	Answer	Time difference td (in seconds)	State before	State after
		t	x	(e)			q	q'
1	000001	09:20:06	ITEM_START	–	–	–	Starting	Stem_Q02
2	000001	09:20:47	ANSWER_SELECTION	Q02	0	41	Stem_Q02	Q03
3	000001	09:20:51	ANSWER_SELECTION	Q03	1	4	Q03	Q01
4	000001	09:20:56	ANSWER_SELECTION	Q01	3	5	Q01	Q02
5	000001	09:21:02	ANSWER_SELECTION	Q02	2	6	Q02	Q04
6	000001	09:21:06	ANSWER_SELECTION	Q04	3	4	Q04	Q02
7	000001	09:21:08	ANSWER_SELECTION		1	2	Q02	Q05
8	000001	09:21:13	ANSWER_SELECTION	Q05	3	5	Q05	…
…	…	…	…	…	…	…	…	Q11
13	000001	09:22:01	ANSWER_SELECTION	Q11	4	4	Q11	Q12
14	000001	09:22:08	ANSWER_SELECTION	Q12	2	7	Q12	Q13
15	000001	09:22:13	ANSWER_SELECTION	Q13	1	5	Q13	Confirmation

Table 3 continued

Line	Raw log data					Augmented information from the FSM Approach		
	Person-identifier	Timestamp	Event name	Question	Answer	Time difference td (in seconds)	State before	State after
		t	x	(e)			q	q'
16	000001	09:22:15	SELECTED_FORWARD	–	–	2	Confirmation	Confirmation
17	000001	09:22:19	MOVE_FORWARD	–	–	4	Confirmation	Endstate

The example illustrates the special case, that a time measures can be extracted for question “Q02”, although question “Q02” was answered as first question. The time measure for the first response “Stem_Q02” is a combination of the time for reading the stem, time for omitting the first answer and the actual time for answering question “Q02”. However, because the value of “Q02” was changed two times, the first event of type ANSWER_SELECTION that belongs to “Q02” is used to identify the transition between state “Starting” and “Stem_Q02” (see line 2) and the second event is used to identify the transition between the state “Q01” and “Q02” (see line 5). Hence, the following measures for the “total time on state *q*” are extracted from the example data as item-level response times using the FSM approach: “Q01” 5 s., “Stem_Q02”: 41 s., “Q02”: 8 s. (line 5 and line 7), “Q03”: 4 s., “Q04”: 4 s., “Q05”: 5 s., “Q11”: 4 s., “Q12”: 7 s., “Q13”: 5 s

The guard “*is_last_event* (ANSWER_SELECTION)” evaluates to true when the log event in line 15, 16 and 17 are processed by the FSM, because none of the following events are of type ANSWER_SELECTION

The guard “*nearest_event_is* (ANSWER_SELECTION, Q…)” is used with respect to a potential value of the column Questions (i.e.,” Q01”, “Q02”, … “Q13”). The guard evaluates to true when the event of type ANSWER_SELECTION for the selected question is the next answer-change event that follows subsequent to a particular event. For the FSM processing the event in line 2, *nearest_event_is*(ANSWER_SELECTION, Q01) evaluates to false, because the next event of type ANSWER_SELECTION is not “Q02”. Instead, *nearest_event_is*(ANSWER_SELECTION, Q03) evaluates to true for the event in line 2. The guard *nearest_event_is*(ANSWER_SELECTION, Q01) evaluates to true for the event in line 3, because the next event of type ANSWER_SELECTION that follows in the event in line 3 belongs to “Q01”

Answers categories: 0 = never or hardly never, 1 = once or twice a month, 2 = once or twice a week, 3 = almost every day, 4 = every day

5.2.2 FSM

The FSM shown in Fig. 6 was used to create an augmented list of tuples for each test taker that contained q' as the state after each log event was processed, together with the time difference td to the previous event. For that purpose we analyzed the XML with an implementation of a generic FSM and extracted the indicators using R (R Core Team 2016). Event-specific information from (e), especially, the name of the question (‘Q01’, …, ‘Q13’) to which an ‘ANSWER_SELECT’-event belongs to, were used in the FSM for the two guards, as described above (see Table 3 for an annotated example). Consequently, the FSM moved from the start s_0 to the state ‘Stem & Q01’ for test takers which choose question ‘Q01’ as the first question and the FSM moved from the state s_0 to state ‘Stem & Q02’ for test takers which started with ‘Q02’ et cetera. Performing such a look-ahead in the FSM allowed us to identify states with a clear meaning and allowed to fit the example into the generic framework.[7]

Item-level response times were extracted from the state summary table. Of all possible states that contained the time for reading the question stem, each test taker visited only one (this directly followed from the definition of the FSM). The state ‘Q01’ that corresponds to answering question ‘Q01’ without reading the stem was only observed for test takers that started with a question different from ‘Q01’ (otherwise the state machine would have moved from the state s_0 to the state ‘Stem & Q01’).

5.3 Results

The left part of Fig. 7 presents a plotted adjacency matrix of all transitions between states created from the state transition table. Obviously, most test takers answered the questions in an ascending order, as the majority of transitions relate start state ‘Starting’ with ‘Stem_Q01’, the state ‘Stem_Q01’ with ‘Q02’, and so on. This pattern is also reflected in the missing value frequencies (see right part of Fig. 7). The time component ‘Q01’, which is only observed for test takers answering question ‘Q01’ not as the first question, is missing by 87%.

Only 55 test taker started with question ‘Q03’, 24 test taker started with the last question and between 2 and 15 test takers started with any of the remaining questions. The total time on states, i.e., the extracted time measures at the item-level, are shown in Fig. 8. Time measures for reading the question stem and answering the first, second or third question (i.e., ‘Stem_Q01’, ‘Stem_Q02’ or ‘Stem_Q03’) are increasing, suggesting that the decision to omit one question (‘Stem_Q02’) or two questions (‘Stem_Q03’) may require time. This clear pattern is not continued for the remaining states that represent reading the stem and answering a subsequent question (‘Stem_Q04’, … ‘Stem_Q13’), likely due to the small frequencies. Note, that this interpretation should also incorporate the items’ content (see Appendix).

[7] Note that an implementation of the state machine approach without look ahead would be possible, requiring to focus on the previous state q from the reconstructed sequence of states in the augmented log data table.

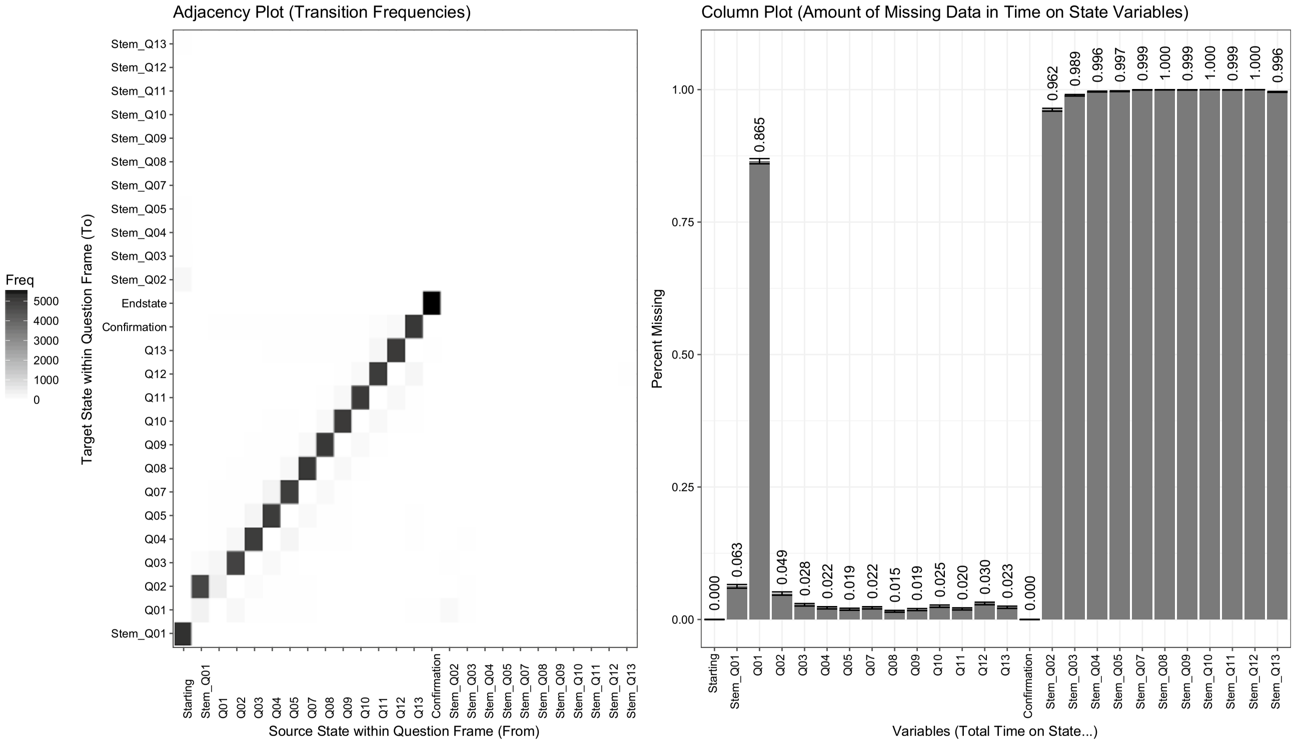


Fig. 7 Adjacency matrix of transitions between states and percent missing values for total time on state the variables

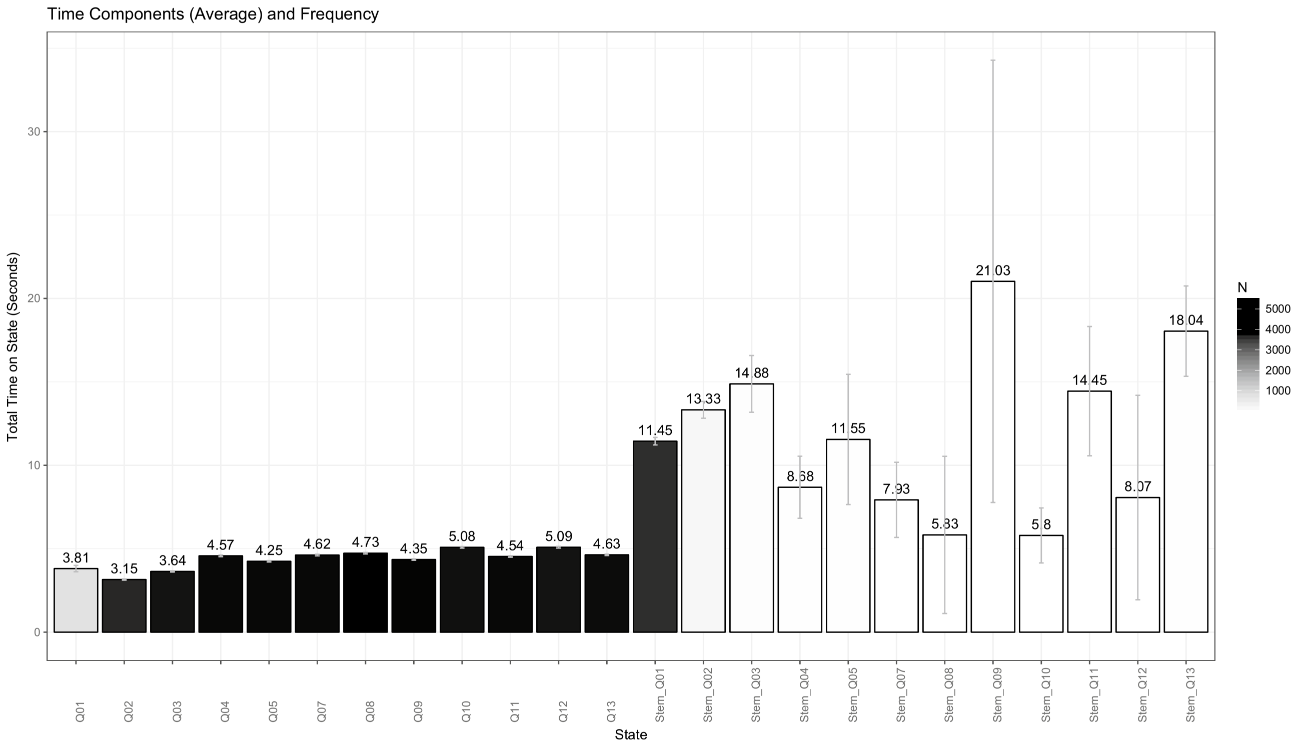


Fig. 8 Average of time components (total time on state) and frequency of item-level response times for the analyzed item

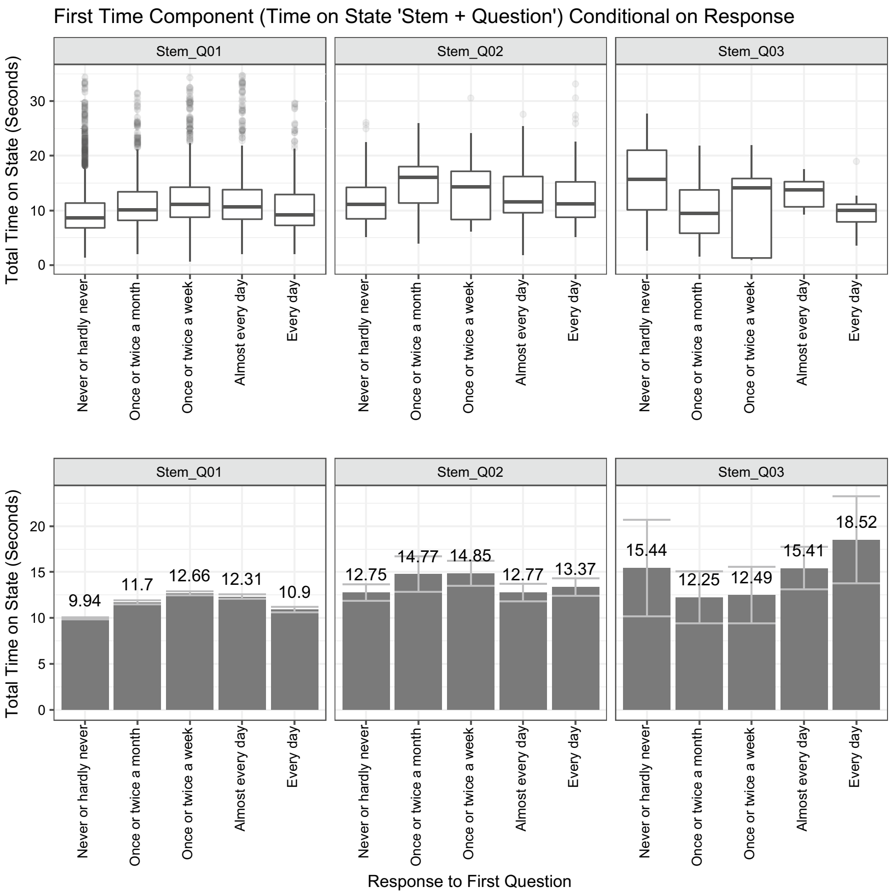


Fig. 9 Total time on state "Stem_Q01", "Stem_Q02" and "Stem_Q3" conditional the response to the first question Q01

As Fig. 8 reveals, the time components for answering the questions 'Q01'…'Q13' range between 3.15 and 5.09 s.

To explore whether the extracted measures for the 'time on state' contain information about the measured construct, we investigate the relationship between this item-level response time and the item response: Fig. 9 summarizes the time on states 'Stem_Q01', 'Stem_Q02' and 'Stem_Q03' conditional on the response to the first question (in addition to box plots in the upper part, means and standard errors are shown in the lower part of Figs. 9 and 10).[8] In particular, for test takers starting with 'Q01' (state 'Stem_Q01') an inverse, u-shaped relationship between response and time component can be observed. For the state 'Stem_Q01' an analysis of variance comparing the effect of the response categories 'Never or hardly never', 'Once or twice a month', 'Once or twice a week', 'Almost every day' and 'Every day' on the

[8] Note, that we set all times larger than 120 s to NA prior to the computation of means and standard errors and the analysis of variance.

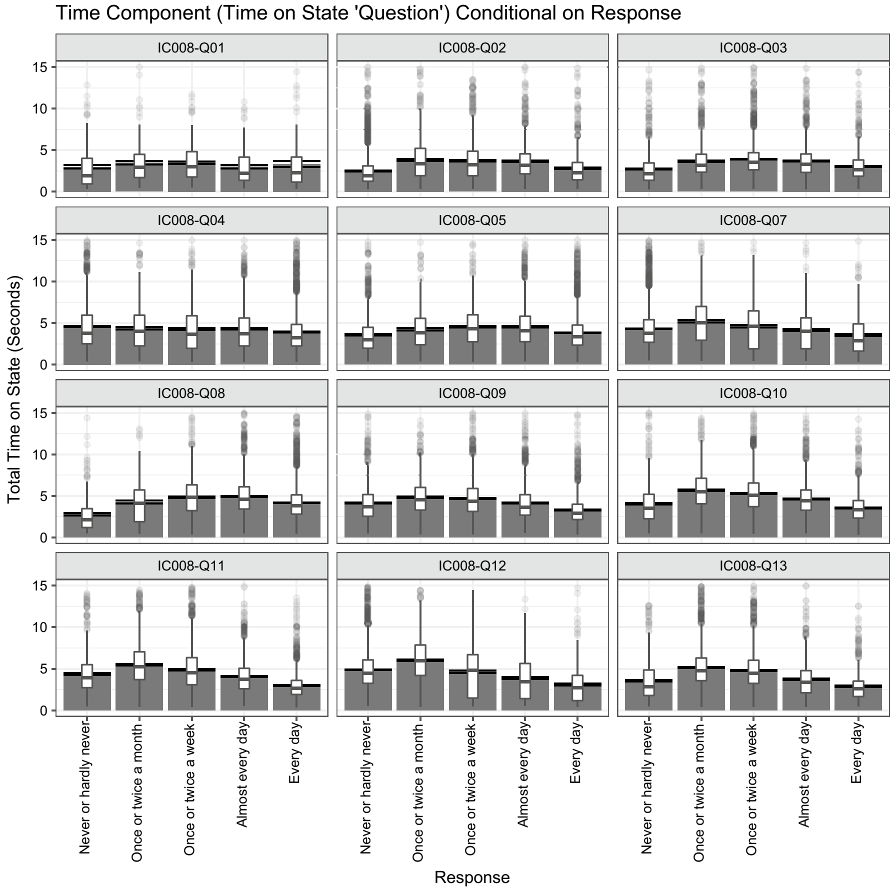


Fig. 10 Total time on state "Q01", "Q02", …, "Q13" conditional the response the question (item-level response after

time revealed significant differences overall $F(4{,}5176) = 47.26$ ($p < 0.01$), representing a small effect ($\eta^2 = 0.04$).

A similar trend can be observed for the item-level response times 'Q01',…, 'Q13' for separate question of the item battery (see Fig. 10). For all states except state 'Q01' we found significant differences for the comparison of mean time measures between the response categories. Moreover, for most states (i.e., all beside 'Q01' and 'Q04') the smallest response time was found in one of the two extreme categories ('Never or hardly never' or 'Every day') together with an inverse, u-shaped response time distribution (skewed for some questions, e.g., 'Q13').

Following the approach that was chosen by Akrami et al. (2007) to investigate the non-linear association between response and response times, we use polynomial regressions with sample weights to support the interpretation of an inverse u-shaped relationship. For that purpose, we treated the response category as a numeric

Table 4 Results of the polynomial regression analyses with selected answer as independent variable and trimmed response

Question	Trend			R^2 total[a]
	Linear R^2	Quadratic ΔR^2	Cubic ΔR^2	
IC008-Q01	**0.025**	**0.026**	0.000	**0.051**
IC008-Q02	**0.020**	**0.055**	**0.001**	**0.076**
IC008-Q03	**0.002**	**0.056**	0.000	**0.059**
IC008-Q04	0.010	0.000	0.001	0.011
IC008-Q05	**0.001**	0.022	0.000	**0.023**
IC008-Q07	**0.002**	**0.013**	**0.001**	**0.017**
IC008-Q08	**0.002**	**0.055**	0.000	**0.057**
IC008-Q09	**0.033**	**0.034**	**0.002**	**0.069**
IC008-Q10	**0.012**	**0.072**	**0.006**	**0.091**
IC008-Q11	**0.081**	**0.046**	**0.005**	**0.133**
IC008-Q12	**0.035**	**0.029**	**0.013**	**0.077**
IC008-Q13	**0.030**	**0.075**	**0.015**	**0.121**

Values in boldface are significant at $p < 0.05$, at least

[a] All trends including insignificant ones

variable, trimmed response times larger than the 95% percentile and estimated and interpreted the R^2 differences between item-level models including a linear, a quadratic and a cubic effect for the response time.

As Table 4 reveals, only for two questions (‘Q04’ and ‘Q05’), the quadratic component was not significantly contributing to the explained variance (quadratic ΔR^2) and for only one question (‘Q04’) the response time did not explain variance in the responses at all (total R^2). Even though for the remaining questions the amount of variance was overall small (the selected response explained between 1.7% for Q07 and 13.2% for ‘Q11’ of response time variance, see column total R^2 in Table 4), the quadratic effect was found to be substantial.

5.4 Summary

Item-level response times for item batteries with multiple questions on one computer screen were found to be ill-defined in the literature and not provided, for instance, in the public use files of the PISA 2015 database. Instead of using the *total time on page* (process-related paradata from subcategory ‘micro’) or the *completion time* for the whole questionnaire (process-related paradata from subcategory ‘macro’) we extracted item-level time measures using multiple log events and a decomposition of the response process into states. The empirical example demonstrates that the FSM approach can be applied directly, and adds to the available tools for the analysis of log data and response times.

In our analysis, the extracted response times were nonlinearly related to the responses. The inverted u-shaped relationship means that responses indicating very high (‘every day’) or low (‘never’) frequency of ICT activities take less time. Probably, for fast responders, the respective item content was clearly congruent or incongruent to their ICT activities. This result pattern, previously found mainly for personality items, is in line with the distance-difficulty hypothesis predicting shorter response times for greater differences between item difficulty and trait level. Note, that systematic relationships between responses and response times as the one found in this application could also be exploited to increase the measurement efficiency of trait measures (see, e.g., Ranger and Ortner 2011).

Using the FSM approach, further research is possible to study the usefulness of the derived time measures, to provide evidence about the validity of their interpretation and to investigate, for instance, their potential to increase measurement precision of PISA context questionnaires. The main purpose of using the FSM approach in this empirical example was to overcome the limitations of post hoc defined indicators, such as the simple time difference used by Zhang and Conrad (2013) and to provide a method to extract item-level response times for item batteries.

The empirical example was simplified by selecting an item screen which was not offering additional features of the platform used in PISA 2015: neither could additional information be requested (button ‘Help’), nor was a consistency check accessible on the selected question screen. Further extensions of the FSM are necessary to apply the suggested procedure to all item batteries of the PISA context questionnaire. As discussed below, this would be much easier if the log data of future PISA would fulfill one of the completeness conditions.

6 Conclusion and discussion

In this paper, we presented a taxonomy integrating different paradata typically gathered in the field of survey research and log data used in educational assessments. Then, a framework was presented that, in particular, elaborates on the extraction of indicators of response- and process-related paradata using FSMs for the retrospective reconstruction of a sequence of states. Both parts are discussed in the following with respect to (i) the c*ompleteness of log data*, (ii) the v*erification of log data* and (iii) the *reproducibility of log data analyses.* Subsequently, limitations and further research will be discussed.

6.1 Completeness of log data

The taxonomy can be used to guide instrument developers in planning upcoming assessments regarding the selection of log data that are relevant for the intended use of the assessment. Beyond that, the taxonomy pointed to the relationship of different data-sources that allows to derive a first set of conditions for defining the completeness of log data. As the collection of log data might impact the performance of an assessment platform, a deliberate selection of all required, but no unnecessary log data is relevant for testing programs and empirical research. In the following, we describe conditions that can be used to determine the completeness of log data. These conditions can be used in test practice for informed decisions about the collection of paradata.

6.1.1 Response completeness

If the answer changes are completely included in the response-related paradata (subcategory ‘answer’), the substantive data can be derived from the log data (*response-completeness*). This condition might require additional information to be included in the log data, for instance, about missing value labels typically used in educational assessments. If a specific assessment platform fulfills the requirement of response completeness, data cleaning procedures and procedures required to guarantee anonymity of the gathered data could be applied exclusively to the log data, because the substantive data can be extracted completely from the log data. Ensuring response completeness is expected to be of great value for test practices when it contributes to a reduction of the effort needed to process log data separately from the result data.

6.1.2 Progress completeness

If timestamps t are available (i.e., if all answer-change event are not only ordered but also allow to reconstruct the substantive data at any particular time), *progress-completeness* is fulfilled. Log data that fulfill this requirement allow, for instance, to apply posterior time limits (e.g., Partchev et al. 2013). This means, the reconstruction of substantive data would only include responses that were given prior to the (item specific) time thresholds RT_i (e.g., Goldhammer and Kroehne 2014). Note, progress-completeness requires timestamps being available for all incremental differences of the status before and after a particular answer-change event (see, for instance, Almond et al. 2012, for a discussion of technical challenges in capturing all keystroke events). Progress-completeness is mainly of interest for psychometric research, for instance, on the relationship between response times and trait as shown in the empirical example.

6.1.3 Replay-completeness

Computer-based tests may be administered in web browsers on heterogeneous devices in unstandardized online assessments (Kroehne et al. 2018). The available display size in terms of pixels and inch will differ very likely between various devices. *Replay-completeness* will be fulfilled, if all information is contained in the log data required to recreate a hypothetical ‘screen cast’ (i.e., representing the progression of visual information presented on screen during the whole session). Replay-completeness is achieved more easily, when the assessment is administered using identical hardware and scrolling is consequently avoided (Dadey et al. 2018). Replay-completeness implies process-completeness under the assumption that all entered responses are presented visually by the platform, and process-completeness implies response-completeness.

6.1.4 State-completeness

The FSM approach contributes to the taxonomy as it shifts the focus from collecting *complete* paradata to the importance of collecting *relevant* log events (e.g., relevant for addressing a certain research question, cf. Goldhammer and Zehner 2017). The relevance of log events depends on the set of states that are supposed to be differentiated by FSMs, because the sequence of states can only be recreated retrospectively, if and only if, the states of a particular machine can be distinguished by observed log events. *State-completeness*, always defined with respect to a FSM or a set of states S, respectively, requires to have all log events available that are needed to identify the transitions defined in the partial state-transition function. When the set of states S becomes part of a study-specific framework that describes how log data are planned to be used for further analysis, the condition of state-completeness can guide instrument developers to formulate requirements with respect to the assessment platform. The FSM shown in Fig. 4 from the empirical application can be understood as an example for missing state-completeness with respect to the specific state 'List of Items'.

State-completeness complements response- and progress-completeness, as it allows to judge which indicators can be computed from the reconstructed sequence of states. However, as the strength of the property state-completeness depends on the substantive meaning of the set S, no direct implications can be deduced regarding the relationship of state-completeness and the completeness conditions derived from the taxonomy of paradata. However, the property of state-completeness is crucial regarding the potential value of log data analyses for psychometric research as well as practical applications. As soon as states of interest can be identified using available log events, methods developed in the field of educational data mining can be used.

6.2 Verification of log data

Two of the completeness conditions provide possibilities to verify log data required for data cleaning and editing of log data.

6.2.1 Implications of response-completeness

Response-completeness can be used to verify existing log data, if the substantive data are stored and successfully cleaned independently from the log data. Existing substantive-data can be compared to the substantive data extracted from the log data, and a perfect match between both representations of the substantive data is expected, whereas differences can be used to identify potential issues that should be resolved before interpreting the data.

6.2.2 Implications of state-completeness

State-completeness allows to use FSMs to verify the match of log events obtained from a platform and the expected transitions, formulated in the state-transition function, allowing to identify both: unexpected log events that might indicate misbehavior of the platform and conceptual errors that resulted in an incomplete state-transition function. The log data are valid with respect to a state machine M, if M accepts log events and all reaches an end state.[9]

6.3 Reproducibility of log data analyses

Indicators in log data analyses are prone to be constructed ad hoc from the available, sometimes severely limited log data stored by a particular platform. Until now, standards for the storage of log data have not been developed, or the standardization to user-specific keys and values is restricted (Hao et al. 2016), impeding the comparability of approaches and threatening the generalizability and validity of results obtained from log data analyses. An important feature of the FSM approach is the possibility to define indicators such as the 'average frequency of clicks' (e.g., Greiff et al. 2016) or the 'number of (relevant) page visits' (e.g., Hahnel et al. 2016) without knowledge about the assessment platform or empirical data. Hence, we hope that the presented approach adds to the tool-kit of methods available for the analysis of log data. Even though state machines will not cover all possible approaches of log file analyses and (educational) data mining, they might become a versatile tool for the feature extraction step (see, e.g., Mislevy et al. 2012).

6.4 Limitations and further research

The analysis of log data from technology-based assessments is still in the fledgling stages and the taxonomy as well as the framework described in this paper are limited in many respects, requiring further research and empirical applications, as discussed in the following.

6.4.1 Taxonomy of paradata

Although the distinction in access-, response- and process-related paradata is apparently useful for instrument developers, the description of subcategories by enumerating prototypical examples is certainly not complete and cannot present a complete overview about the literature. For instance, sensor data such as eye-tracking, heart

[9] Technically, the underlying idea is, that the FSM will either accept an input $x \in \sum$, if a transition is defined for the current state $q \in S$ (i.e., the partial state-transition function $\delta(q, x)$ returns a new state q') or reject the input, if for a particular input element $x \in \Sigma$ no transition is defined for $q \in S$. Note that if input elements (i.e., log events) should be ignored for a particular application of a state machine to analyze log data, the partial state-transition function $\delta(q, x)$ can contain transitions from state q to the identical state q for those input elements $x \in \sum$.

rate and motion tracking data are not included in the taxonomy, calling for another extension in the upcoming research. An additional limitation is, that the taxonomy does not cover the needed meta-data for log data. Further research might hook up with defining meta-data for the different components of the presented taxonomy.

6.4.2 Finite state machine approach

The FSM approach should facilitate theory-guided data mining by requiring the definition and enumeration of meaningful states that represent distinct parts of the interplay between test taker and assessment platform. The choice of the formalization using finite state machines seems to some extent arbitrary, because extensions such as guards, variables and orthogonal regions and, in particular, look-ahead operations that are desirable for practical applications go beyond the formal definition of state machines presented in this paper. However, key aspects, i.e., the differentiation between a finite number of states that can be described theoretically, and log events that are used to identify transitions to retrospectively reconstruct the sequence of states can be applied also with extended state machines.

Further applications of the framework might incorporate formal properties of the more powerful state machines that include hierarchically nested states and orthogonal regions (see, e.g., Alagar and Periyasamy 2011). Integrating multiple machines (i.e., orthogonal regions) is expected to allow the recognition of more complex behavioral patterns. For instance, if computer-based instruments are designed to distinguish different solution behaviors of test takers (e.g., exploration of a system versus application of the knowledge about a system; see, e.g., Tóth et al. 2014) and if log events can be used to identify transitions between corresponding states, the classification of solution behavior with finite state machines can help to increase the diagnostic usage of log data.

Further applications are needed to show, whether the approach can help to simplify the analysis of log data while strengthening the relationship to domain knowledge and assessment frameworks at the same time.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Appendix

The question text for IC008 was ‘How often do you use digital devices for the following activities outside of school?’ Students were instructed to select ‘*Never or hardly never*’, ‘*Once or twice a month*’, ‘*Once or twice a week*’, ‘*Almost every day*’ or ‘*Every day*’ to each of the following activities:

Q01: Playing one-player games.

Q02: Playing collaborative online games.

Q03: Using email.

Q04: <Chatting online> (e.g., <MSN®>).

Q05: Participating in social networks (e.g., <Facebook>, <MySpace>).

Q07: Playing online games via social networks (e.g., <Farmville®>, <The Sims Social>).

Q08: Browsing the Internet for fun (such as watching videos, e.g., <YouTube™>).

Q09: Reading news on the Internet (e.g., current affairs).

Q10: Obtaining practical information from the Internet (e.g., locations, dates of events).

Q11: Downloading music, films, games or software from the internet.

Q12: Uploading your own created contents for sharing (e.g., music, poetry, videos, computer programs).

Q13: Downloading new apps on a mobile device.

References

Akrami N, Hedlund LE, Ekehammar B (2007) Personality scale response latencies as self-schema indicators: the inverted-U effect revisited. Pers Individ Differ 43(3):611–618

Alagar VS, Periyasamy K (2011) Specification of software systems, 2nd edn. Springer, New York

Almond R, Deane P, Quinlan T, Wagner M, Sydorenko T (2012) A preliminary analysis of keystroke log data from a timed writing task (research report 12–23). Educational Testing Service, Princeton

Bennett RE (2015) The changing nature of educational assessment. Rev Res Educ 39(1):370–407

Bergner Y, Shu Z, Von Davier AA (2014) Visualization and confirmatory clustering of sequence data from a simulation-based assessment task. In: Proceedings of the 7th international conference on educational data mining (EDM 2014), pp 177–184

Bridgeman B, Lennon ML, Jackenthal A (2003) Effects of screen size, screen resolution, and display rate on computer-based test performance. Appl Meas Educ 16(3):191–205

Callegaro M (2012) A taxonomy of paradata for web surveys and computer assisted self interviewing. In: Poster presented at the general online research conference, Mannheim, Germany, March 2012

Couper M (1998) Measuring survey quality in a CASIC environment. In: Proceedings of the section on survey research methods of the American Statistical Association, pp 41–49

Couper MP, Kreuter F (2013) Using paradata to explore item level response times in surveys. J R Stat S Ser A 176(1):271–286

Couper MP, Tourangeau R, Conrad FG, Zhang C (2013) The design of grids in web surveys. Soc Sci Comput Rev 31(3):322–345

Dadey N, Lyons S, DePascale C (2018) The comparability of scores from different digital devices: a literature review and synthesis with recommendations for practice. Appl Meas Educ 31(1):30–50. https://doi.org/10.1080/08957347.2017.1391262

Durrant GB, D’Arrigo J, Steele F (2011) Using paradata to predict best times of contact, conditioning on household and interviewer influences. J R Stat Soc Ser A 174(4):1029–1049

Eisenberg P, Wesman AG (1941) Consistency in response and logical interpretation of psychoneurotic inventory items. J Educ Psychol 32(5):321–338

Ferrando PJ, Lorenzo-Seva U (2007) A measurement model for Likert responses that incorporates response time. Multivar Behav Res 42(4):675–706

Ferreira DR (2017) A primer on process mining. Springer, Cham

Gabadinho A, Ritschard G, Mueller NS, Studer M (2011) Analyzing and visualizing state sequences in R with TraMineR. J Stat Softw 40(4):1–37

Goldhammer F, Kroehne U (2014) Controlling individuals' time spent on task in speeded performance measures: experimental time limits, posterior time limits, and response time modeling. Appl Psychol Meas 38(4):255–267

Goldhammer F, Zehner F (2017) What to make of and how to interpret process data. Meas Interdiscip Res Perspect 15(3–4):128–132

Goldhammer F, Naumann J, Stelter A, Tóth K, Roelke H, Klieme E (2014) The time on task effect in reading and problem solving is moderated by task difficulty and skill: insights from a computer-based large-scale assessment. J Educ Psychol 106(3):608–626

Greiff S, Niepel C, Scherer R, Martin R (2016) Understanding students' performance in a computer-based assessment of complex problem solving: an analysis of behavioral data from computer-generated log files. Comput Hum Behav 61:36–46

Hahnel C, Goldhammer F, Naumann J, Kroehne U (2016) Effects of linear reading, basic computer skills, evaluating online information, and navigation on reading digital text. Comput Hum Behav 55:486–500

Hanly M, Clarke P, Steele F (2016) Sequence analysis of call record data: exploring the role of different cost settings. J R Stat Soc Ser A 179(3):793–808

Hao J, Shu Z, von Davier A (2015) Analyzing process data from game/scenario-based tasks: an edit distance approach. J Educ Data Min 7(1):33–50

Hao J, Smith L, Mislevy R, von Davier A, Bauer M (2016) Taming log files from game and simulation based assessment: data model and data analysis tool (research report 16–10). Educational Testing Service, Princeton

He Q, von Davier M (2015) Identifying feature sequences from process data in problem-solving items with n-grams. In: van der Ark LA, Bolt DM, Wang WC, Douglas JA, Chow SM (eds) Quantitative psychology research. Springer International Publishing, Cham, pp 173–190

He Q, von Davier M (2016) Analyzing process data from problem-solving items with n-grams: insights from a computer-based large-scale assessment. Handbook of research on technology tools for real-world skill development. IGI Global, Hershey, pp 750–777

Heerwegh D (2003) Explaining response latencies and changing answers using client-side paradata from a web survey. Soc Sc Comput Rev 21(3):360–373

Higgins J, Russell M, Hoffmann T (2005) Examining the effect of computer-based passage presentation of reading test performance. J Technol Learn Assess 3:1–36

Höhne JK, Schlosser S (2018) Investigating the adequacy of response time outlier definitions in computer-based web surveys using paradata surveyfocus. Soc Sci Comput Rev 36(3):369–378

IMS Global Learning Consortium (2012) IMS question and test interoperability assessment test, question and item information. https://www.imsglobal.org/question/qtiv2p1/imsqti_infov2p1.html. Accessed 22 Feb 2018

Jude N (2016) The assessment of learning contexts in PISA. In: Kuger S, Klieme E, Jude N, Kaplan D (eds) Assessing contexts of learning. methodology of educational measurement and assessment. Springer, Cham, pp 39–51

Kaczmirek L (2009) Human-survey interaction: usability and nonresponse in online surveys. von Halem, Köln

Khasawneh N, Al-Salman R, Al-Hammouri AT, Conrad S (2012) A generic framework for collecting and mining client paradata for web applications. J Emerg Technol Web Intell 4(4):324–332

Klausch T, Hox JJ, Schouten B (2013) Assessing the mode-dependency of sample selectivity across the survey response process. Statistics Netherlands, The Hague

Kreuter F (2013) Improving surveys with paradata: analytic uses of process information, vol 581. Wiley, Hoboken

Kroehne U, Gnambs T, Goldhammer F (2018) Disentangling setting and mode effects for online competence assessment. In: Blossfeld H-P, Roßbach H-G (eds) Education as a lifelong process. Springer VS, Wiesbaden

Liu M, Cernat A (2016) Item-by-item versus matrix questions: a web survey experiment. Soc Sci Comput Rev. https://doi.org/10.1177/0894439316674459

Luecht RM, Clauser BE (2002) Test models for complex CBT. In: Mills CN (ed) Computer-based testing: building the foundation for future assessments. Erlbaum Associates, Mahwah, pp 67–88

Luecht RM, Sireci SG (2011) A review of models for computer-based testing (research report no. 2011–12). College Board, New York

Ma Y, Baker R, Agnihotri L, Plaza P, Mojarad S (2016) Effect of student ability and question difficulty on duration. In: Proceedings of the 9th international conference on educational data mining, pp 135–142

Malhotra N (2008) Completion time and response order effects in web surveys. Public Opin Q 72:914–934

Mavletova A, Couper MP (2016) Grouping of items in mobile web questionnaires. Field Methods 28(2):170–193

Mayerl J (2013) Response latency measurement in surveys. Detecting strong attitudes and response effects. Surv Methods Insights Field. http://surveyinsights.org/p=1063. Accessed 26 Feb 2018

McClain CA, Couper MP, Hupp AL, Keusch F, Peterson G, Piskorowski AD, West BT (2018) A typology of web survey paradata for assessing total survey error. Soc Sci Comput Rev. https://doi.org/10.1177/0894439318759670

Mislevy RJ, Behrens JT, Dicerbo KE, Levy R (2012) Design and discovery in educational assessment: evidence-centered design, psychometrics, and educational data mining. J Educ Data Min 4(1):11–48

Mislevy RJ, Corrigan S, Oranje A, DiCerbo K, Bauer MI, von Davier A, John M (2016) Psychometrics and game-based assessment. In: Drasgow F (ed) Technology and testing: improving educational and psychological measurement. Routledge, New York, pp 23–48

Molenaar D, Tuerlinckx F, van der Maas HLJ (2015) A bivariate generalized linear item response theory modeling framework to the analysis of responses and response times. Multivar Behav Res 50(1):56–74

Nachtigall C, Kroehne U, Funke F, Steyer R (2003) Pros and cons of structural equation modeling. Methods Psychol Res Online 8(2):1–22

Neubert JC, Kretzschmar A, Wüstenberg S, Greiff S (2015) Extending the assessment of complex problem solving to finite state automata: embracing heterogeneity. Eur J Psychol Assess 31(3):181–194

OECD (2016) PISA 2015 assessment and analytical framework: science, reading, mathematic and financial literacy. PISA, OECD Publishing, Paris

OECD (2017) PISA 2015 technical report. PISA, OECD Publishing, Paris

Olson K (2013) Paradata for nonresponse adjustment. Ann Am Acad Political Soc Sci 645(1):142–170

Olson K, Parkhurst B (2013) Collecting paradata for measurement error evaluations. In: Kreuter F (ed) Improving surveys with paradata. Wiley, Hoboken, pp 43–72

Partchev I, De Boeck P, Steyer R (2013) How much power and speed is measured in this test? Assessment 20(2):242–252

R Core Team (2016) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Ramalingam D, Adams RJ (2018) How can the use of data from computer-delivered assessments improve the measurement of twenty-first century skills? In: Care E, Griffin P, Wilson M (eds) Assessment and teaching of 21st century skills. Springer International Publishing, Cham, pp 225–238

Ranger J, Ortner TM (2011) Assessing personality traits through response latencies using item response theory. Educ Psychol Meas 71(2):389–406

Reips UD (2002) Standards for internet-based experimenting. Exp Psychol (formerly Zeitschrift für Experimentelle Psychologie) 49(4):243–256

Reips UD (2010) Design and formatting in internet-based research. In: Gosling S, Johnson J (eds) Advanced methods for conducting online behavioral research. American Psychological Association, Washington, D.C, pp 29–43

Richter T, Naumann J (2000) Computer-based assessment of reading skills. In: Proceedings of the 2nd computers in psychology conference (CiP 2000). (WWW document). https://pdfs.semanticscholar.org/a692/54b93140997e704e7c65259a8f6021010350.pdf. Accessed 12 Feb 2018

Roelke H (2012) The ItemBuilder: a graphical authoring system for complex item development. In: Proceedings of world conference on E-learning in corporate, government, healthcare, and higher education, Chesapeake, pp 344–353

Romero C (ed) (2011) Handbook of educational data mining. Taylor & Francis, Boca Raton

Scherer R, Greiff S, Hautamäki J (2015) Exploring the relation between time on task and ability in complex problem solving. Intelligence 48:37–50. https://doi.org/10.1016/j.intell.2014.10.003

Schnipke DL, Scrams DJ (2002) Exploring issues of examinee behavior: insights gained from response-time analyses. In: Mills CN, Potenza M, Fremer JJ, Ward W (eds) Computer-based testing: building the foundation for future assessments. Lawrence Erlbaum Associates, Hillsdale, pp 237–266

Schroeders U, Wilhelm O (2010) Testing reasoning ability with handheld computers, notebooks, and paper and pencil. Eur J Psychol Assess 26(4):284–292

Sinharay S, Wan P, Whitaker M, Kim DI, Zhang L, Choi SW (2014) Determining the overall impact of interruptions during online testing. J Educ Meas 51(4):419–440

Stieger S, Reips UD (2010) What are participants doing while filling in an online questionnaire: a paradata collection tool and an empirical study. Comput Hum Behav 26(6):1488–1495

Tóth K, Rölke H, Greiff S, Wüstenberg S (2014) Discovering students' complex problem solving strategies in educational assessment. In: Proceedings of the 7th international conference on educational data mining. International Educational Data Mining Society, pp 225–228

Way WD, Davis LL, Keng L, Strain-Seymour E (2015) From standardization to personalization: the comparability of scores based on different testing conditions, modes, and devices. In: Drasgow F (ed) Technology and testing: improving educational and psychological measurement. Routledge, New York

Wood D, Harms PD, Lowman GH, DeSimone JA (2017) Response speed and response consistency as mutually validating indicators of data quality in online samples. Soc Psychol Pers Sci 8(4):454–464

Yan T, Tourangeau R (2008) Fast times and easy questions: the effects of age, experience and question complexity on web survey response times. Appl Cogn Psychol 22(1):51–68

Zhang C, Conrad FG (2013) Speeding in web surveys: the tendency to answer very fast and its association with straightlining. Surv Res Methods 8(2):127–135

Zoanetti N (2010) Interactive computer based assessment tasks: how problem-solving process data can inform instruction. Australas J Educ Technol 26(5):585–606