Board Mining: Understanding the Use of Board-Based Collaborative Work Management Tools*

Alfonso Bravo[1][0000−0002−6376−0811], Cristina Cabanillas[1,2][0000−0001−9182−8847], Joaquín Peña[2][0000−0001−9216−9695], and Manuel Resinas[1,2][0000−0003−1575−406X]

[1] SCORE Lab, Universidad de Sevilla, Spain
[2] I3US Institute, Universidad de Sevilla, Spain
{abllanos,cristinacabanillas,joaquinp,resinas}@us.es

1 Board Mining: Definitions and operations

We use the term board mining to refer to the extraction of information related to a board based on its event log. To this end, we start with the definition of what a board-based event log is. Then, we discuss about structural updates of a board and how to detect them. After that, we describe how to discover a board-based design model based on the metamodel introduced in [1]. Next, we detail how an board-based event log of a particular kind of boards can be transformed into a process event log. Finally, we report on a set of metrics that can be used to characterize boards and their relationship with board-based design patterns [1].

1.1 Board-based event logs

A board-based event log is a collection of all the events that occur in a board. Each board has its own event log and we consider the event log of one board to be independent of the others. Conceptually, a board-based event log is similar to a business process event log. The main difference is that instead of using the concepts of case and activity, it uses the concepts of list and card. To formally define a board-based event log, we start by defining some universes.

Definition 1 (Universes). *We define the following universes for this paper:*

- $\mathcal{U}_{ei}$ *is the universe of event identifiers,*
- $\mathcal{U}_{list}$ *is the universe of lists,*
- $\mathcal{U}_{card}$ *is the universe of cards,*
- $\mathcal{U}_{time}$ *is the universe of timestamps,*
- $\mathcal{U}_{type} = \{c, x, m, d, a, l_c, l_u, l_x\}$ *is the universe of actions that can be performed in a board, namely: create (c), close (x), move (m), delete (d), and update (u) a card; and create* l_c*, update* l_u*, and close (*l_x*) a list.*

* Work funded by grants RTI2018-100763-J-I00 and RTI2018-101204-B-C22 funded by MCIN/ AEI/ 10.13039/501100011033/ and ERDF A way of making Europe; and grant P18-FR-2895 funded by Junta de Andalucía/FEDER, UE.

– *$\mathcal{U}_{event} = \mathcal{U}_{ei} \times \mathcal{U}_{time} \times \mathcal{U}_{list} \times (\mathcal{U}_{card} \cup \perp) \times \mathcal{U}_{type} \times (\mathcal{U}_{list} \cup \perp)$ is the universe of events.*

$e = (ei, time, l, c, type, lt) \in \mathcal{U}_{event}$ is an event with identifier ei, corresponding to the execution of action $type$ on a card c that is in list l. If $type$ is m (moving a card), then lt is the list to which the card is moved. Otherwise, $lt = \perp$. Also, if $type \in \{l_c, l_u, l_x\}$, i.e. it is a list action, then both $c = \perp$ and $lt = \perp$.

Definition 2 (Event Projection). *Given $e = (ei, time, l, c, type, lto) \in \mathcal{U}_{event}$, $\pi_{ei}(e) = ei$, $\pi_{time}(e) = time$, $\pi_l(e) = l$, $\pi_c(e) = c$, $\pi_{type}(e) = type$, and $\pi_{lto}(e) = lto$.*

A board-based event log is a collection of totally ordered events, where event identifiers are unique.

Definition 3 (Board-based event log). *$(E, \prec_E)$ is a board-based event log with $E \subseteq \mathcal{U}_{event}$ and $\prec_E \subseteq E \times E$ such that:*

- *$\prec_E$ defines a total order,*
- *$\forall_{e1,e2 \in E} \pi_{ei}(e1) = \pi_{ei}(e2) \Rightarrow e1 = e2$, and*
- *$e1 \prec e2 \Rightarrow \forall_{e1,e2 \in E} \pi_{time}(e1) < \pi_{time}(e2)$*

Further attributes could be found in the event log, e.g. the name of the lists and cards, due dates, comments, attachments, checklists. We have not included them because they are not used in this paper. However, they could be added following the same approach that is used to add attributes in process event logs.

For convenience, we also define the filters of the event log by type as follows.

Definition 4 (Filters by type). *Let $(E, \prec_E)$ be a board-based event log, a filter by type of the event log is defined as $\phi_t(E) = \{e \in E | \pi_{type}(e) = t\}$.*

1.2 Discovering structural updates

As described in Section **??**, the structure of a board evolves with its use. We define the structural evolution of a board based on two operations. One involves identifying the timeline of every list in the board (i.e. the moment at which the list is created and the moment at which it is closed), and the other operation involves identifying the moments in which a significant structural update has been performed. The former operation is specified in Def. 5.

Definition 5 (List evolution). *Let $(E, \prec_E)$ be a board-based event log. The list evolution of the board $LE \subseteq \mathcal{U}_{list} \times \mathcal{U}_{time} \times \mathcal{U}_{time}$ assigns a begin time and an end time to each list. It can be defined as $LE = \{(l, t_s, t_e) | \exists_{e_s, e_e \in E} \pi_l(e_s) = \pi_l(e_e) = l \wedge \pi_{type}(e_s) = l_c \wedge \pi_{time}(e_s) = t_s \wedge \pi_{type}(e_e) = l_x \wedge \pi_{time}(e_e) = t_e\} \cup \{(l, t_s, e) | (\nexists_{e_e \in E} \pi_l(e_e) = l \wedge \pi_{type}(e_e) = l_x) \wedge (\exists_{e_s \in E} \pi_l(e_s) = l \wedge \pi_{type}(e_s) = l_c \wedge \pi_{type}(e_s) = t_s) \wedge \exists_{e_e \in E} last(e_e) \wedge \pi_{time}(e_e) = t_e\}$, where $last(e) = \nexists_{e' \in E} e \prec e'$.*

Note that for lists that have not been closed, we take the timestamp of the last event of the board as reference for the last event of the list. It will be the same for all lists of the board that are still open.

The definition of the structural change period is based on three observations. First, we consider that a structural change is potentially triggered by any event related to a list (a list-related event), i.e. creation, update, or closure. Second, in many cases, a structural change is not caused by a single list-related event but by a set of many list-related events. This is particularly common in the structural updates that lead to a redesign of the board. Third, a structural update may not only include list-related events but also card-related events. For instance, after creating a new list, we may want to reorganize the cards on the board. To define a structural change period based on these three observations, we first introduce the concept of structural change events (Def. 6) as the events that are involved in a structural change according to the first and third observations. The latter is materialized by considering a threshold period in which all events are assumed to be part of a structural change. Based on the structural change events, we can define the structural change intervals of a board (Def. 7).

Definition 6 (Structural change events). *Let $(E, \prec_E)$ be a board-based event log, let θ be a time period (e.g., one day), the set of events that are involved in a structural change $SE^\theta \subseteq E$ can be defined as $SE^\theta = \{e \in E | \pi_{type}(e) \in \{l_c, l_u, l_x\} \vee \exists_{e' in E} \pi_{time}(e) - \pi_{time}(e') < \theta$*

Definition 7 (Structural change intervals). *Let $(E, \prec_E)$ be a board-based event log, let SE^θ be the structural change events considering a time period θ, and let ϵ be a real number that represents the minimum number of list-events that need to be contained in the structural change interval. The potential structural change intervals are a set of time intervals $SC^{\theta,\epsilon} = \{(t_s, t_e) | \exists_{e_s, e_e \in SE^\theta} \pi_{time}(e_s) = t_s \wedge \pi_{time}(e_e) = t_e \wedge prev(e_s), next(e_e) \notin SE^\theta \wedge \forall_{e \in E} e_s \prec e \prec e_e \Rightarrow e \in SE^\theta \wedge |\{e \in SE^\theta | e_s \prec e \prec e_e \wedge \pi_{type}(e) \in \{l_c, l_u, l_x\}\}| \geq \epsilon$, where $prev(e)$ (resp. $next(e)$) represents the event that is immediately before (resp. after) e.*

Together, list evolution and structural change intervals provide a useful tool to analyze the evolution of the structure of a board as discussed in Section **??**.

1.3 Discovering a board design model

Apart from structural updates, the use of a board involves creating, updating, moving, deleting and closing cards. Although in general, BBTs give freedom to users to apply any of these actions on any card that is on any list, in practice, the board users set implicit rules about how the board should be used. For instance, let us think of a board with the following lists: *Inbox*, *Planned*, *Doing*, and *Done*. Implicit rules on this board might be that new cards should always be created in *Inbox*. Then, they should be moved to *Planned*, *Doing*, and *Done* in that order. Cards in *Done* should be closed from time to time to keep the board manageable. Finally, cards should be updated (e.g. add new comments or new attachments) while they are on lists *Planned* and *Doing*. The goal of discovering a board design model involves using the information from the BBT log to uncover these rules. To this end, we first need to define a board design metamodel. This metamodel is based on the one presented in [1] but with two key differences. First, we do not distinguish between lists that contain information cards and lists that contain task cards because performing this distinction automatically is out of the scope of this paper. Second, we include additional elements that reflect the interaction of the user with the cards other than their movement between lists.

Definition 8 (Board design). *A board design is a tuple* $B = (L, cf, sp, cc, cx, cu)$*, where:*

- $L \in \mathcal{U}_{list}$ *is the set of lists of the board,*
- $cf \subseteq \mathcal{P}(L)^3$ *is the definition of the card flow of the board, i.e., the set of lists between which the cards can move,*
- $sp \subseteq L \times L$ *represent the semantic precedence between lists. When we say that there is semantic precedence between two lists in a board we are specifying that there is some high-level connection between them, besides their visual representation,*
- $cc \subseteq L$ *is the set of lists in which cards are created,*
- $cx \subseteq L$ *is the set of lists in which cards are closed,*
- $cu \subseteq L$ *is the set of lists in which cards are updated.*

Two considerations must be made about board designs. First, note that we are representing a board design (i.e. its structure and the implicit rules that determine how it should be used), not an actual board use. That is the reason why cards are not part of a board design. Second, this definition does not provide information on how the structure of the board can change. For instance, in some board designs the creation of a list is part of the normal behavior of the board (e.g. a new list created for each month in a publication calendar). The analysis of such predetermined structural changes is out of the scope of this paper.

Intuitively, using a BBT log, it is possible to discover a board design as follows. The lists are the set of lists that appear in the events of the log. The card flow is the set of lists that are connected to each other by means of card moves. The set of lists created, closed and updated are composed of those lists where cards are created, updated or closed with a frequency higher than a threshold.

Definition 9 (Connected lists). *Let* $(E, \prec_E)$ *be a board-based event log, and let* $G = (N, E)$ *be a directed graph where the nodes are the lists of the event log* $N = \{l \in \mathcal{U}_{list} | \exists_{e \in E} \pi_l(e) = l\}$ *and the edges are the list pairs* $(l1, l2)$ *with* $l1 \neq l2$ *that represent that a card has been moved from* $l1$ *to* $l2$*:* $E = \{(l1, l2) \in N \times N | \exists_{e \in \phi_m(E)} \pi_l(e) = l1 \wedge \pi_{lto}(e) = l2\}$*. The set of connected lists* $C_l \subseteq \mathcal{U}_{list}$ *is the set of connected components of the graph* G*.*

Definition 10 (Board design discovery). *Let* $(E, \prec_E)$ *be a board-based event log, and let* λ *be a real number. A board design* $B = (L, cf, sp, cc, cx, cu)$ *can be discovered from the event log as follows:*

[3] $\mathcal{P}(L)$ is the powerset of the lists of the board

- $L = \{l \in \mathcal{U}_{list} | \exists_{e \in E} \pi_l(e) = l\}$
- $cf = C_l$
- $cc = \{l \in L | \frac{|\{e \in \phi_c(E) | \pi_l(e) = l\}|}{|\phi_c(E)|}\} > \lambda$
- $cx = \{l \in L | \frac{|\{e \in \phi_x(E) | \pi_l(e) = l\}|}{|\phi_x(E)|}\} > \lambda$
- $cu = \{l \in L | \frac{|\{e \in \phi_u(E) | \pi_l(e) = l\}|}{|\phi_u(E)|}\} > \lambda$

Where L is the set of lists, cf is the card flow, and cc, cx, and cu are the lists where cards are created, closed or updated with a frequency higher than the threshold.

Notice that although we have established one shared threshold for creation, update and closure, it is straightforward to extend the definition to include one threshold for each of them.

Semantic precedence has not been included in Def. 10 because it requires an additional consideration. A semantic precedence between two lists involves a high-level relationship between them. However, the way this high-level relationship manifests can change depending on the semantics of the board [1]. For instance, in a board where lists represent a lifecycle and cards move through these lists, the semantic precedence manifests by card movement. On the contrary, in another board, lists may represent the months of the year and cards may not move through these lists. In this case, there is a semantic precedence between lists (i.e. the temporal precedence) but it manifests differently, e.g. by periods of card creations or updates that occur in a different list every month. In this paper, we focus on the semantic precedence that manifests by card movement. Specifically, we consider there is a semantic precedence between list $l1$ and $l2$ if the number of cards that move from $l1$ to $l2$ is higher than a threshold.

Definition 11 (Flow-based semantic precedence). *Let $(E, \prec_E)$ be a board-based event log, and λ be a real number between 0 and 1. The flow-based semantic precedence of the log is $sp = \{(l1, l2) \in L \times L | \frac{|\{e \in \phi_m(E) | \pi_l(e) = l1 \wedge \pi_{lto}(e) = l2\}|}{|\phi_m(E)|} > \lambda\}$*

1.4 Creating a process event log

A consequence of the discussion about semantic precedence is that depending on the board and the semantics associated to its lists and cards, it is possible to identify a process event log derived from the board-based event log.

Definition 12 (Process event log). *Let $\mathcal{U}_{act}$ be the universe of activity names, $\mathcal{U}_{ci}$ the universe of case identifiers, and $\mathcal{U}_{pevent} = \mathcal{U}_{ei} \times \mathcal{U}_{act} \times \mathcal{U}_{ci} \times \mathcal{U}_{time}$ is the universe of process events. A process event log is a tuple $(E_p, \prec_{E_p})$ with $E_p \subseteq \mathcal{U}_{pevent}$ and $\prec_{E_p} \subseteq E_p \times E_p$ a set that defines a total order between the events in E_p with the same conditions as in Definition 3.*

In this paper, we focus on boards that follow the *information lifecycle* or *kanban* patterns identified in [1]. In them, cards may represent tasks or resources; they are moved around lists, which represent activities or stages; and there is a semantic precedence between the lists based on card movement. The mapping to a process event log is thus rather straightforward since cards can be seen as cases, and lists can be seen as activities. In fact, the flow-based semantic precedence defined before is exactly the directly-follows graph that can be obtained from this mapping with a frequency threshold for edges.

Definition 13 (Board-based event log to process event log). *Let* $(E, \prec_E)$ *be a board-based event log of a board that follows the information lifecycle or kanban patterns, and let* $E^c = \{e \in E | \pi_{type}(e) \in \{c, m, x\}\}$ *be the subset of events in* E *that involve creation, move or closure of cards. A mapping function* $\mu(E, \prec_E) = (E_p, \prec_{E_p})$ *can be defined to transform the board-based event log into a process event log such that* $E_p = \{(ei, act, ci, time) | \exists_{e \in E^c} \pi_{ei}(e) = ei \wedge \pi_{time}(e) = time \wedge \pi_c(e) = ci \wedge map(e) = act\}$*, where*

$$map(e) = \begin{cases} created, & if\ \pi_{type}(e) = c \\ \pi_l(e), & if\ \pi_{type}(e) = m \\ \pi_l(e), & if\ \pi_{type}(e) = x \end{cases}$$

$$\prec_{E_p} = \{(e_{p1}), e_{p2}) \in E_p \times E_p | \exists_{(e_1, e_2) \in E^c} e_1 \prec e_2 \wedge \mu(e_1) = e_{p1} \wedge \mu(e_2) = e_{p2}.$$

There are two considerations about this mapping. First, if we had considered additional data attributes in the board-based event log, their mapping to a process event log would be a straightforward one-to-one mapping. Second, we are assuming that the process events correspond to completed activities. For this reason, each activity will be added in the process log when the card is moved out of the respective list or it is closed. We also add an initial activity called *created* to represent the creation event. Alternatives for mapping this are to consider that the process events represent the beginning of an activity, or include a lifecycle transition attribute in the process event log to note both start and end events.

Although as detailed in Section **??** and also discussed in [1], lifecycle and kanban boards represent a significant part of the boards analyzed, there are other possible semantics for boards (e.g. consider the example of the lists that represent months) that require other mappings.

1.5 Board metrics

Based on the operations described in the previous sections, it is possible to define a set of metrics that can be used to characterize the use of a board based on the concept of card flow. In Section **??**, we detail how these metrics can been used to characterize the boards in a dataset of BBT logs.

Definition 14 (Board use metrics). *Let* $(E, \prec_E)$ *be a board-based event log, and* C_l *the set of connected lists of* E*, the following metrics based on the actions performed on cards:*

- $\#\ lists = |\{l \in \mathcal{U}_{list} | \exists_{e \in E} \pi_l(e) = l\}|$
- $\#\ cards = |\{c \in \mathcal{U}_{card} | \exists_{e \in E} \pi_c(e) = c\}|$

- $\#$ *moving cards* $= |\{c \in \mathcal{U}_{card} | \exists_{e \in \phi_m(E)} \pi_c(e) = c\}|$
- $\#$ *closed cards* $= |\{c \in \mathcal{U}_{card} | \exists_{e \in \phi_x(E)} \pi_c(e) = c\}|$
- *Relative* $\#$ *moving cards* $= \frac{\#\ moving\ cards}{\#\ cards}$
- *Relative* $\#$ *closed cards* $= \frac{\#\ closed\ cards}{\#\ cards}$
- *Relative* $\#$ *lists with card movement* $= \frac{|\{l \in \mathcal{U}_{list} | \exists_{e \in \phi_m(E)} (\pi_l(e) = l \vee \pi_{lto}(e) = l)\}|}{\#\ lists}$
- *Relative* $\#$ *moves per moving card* $= \frac{\frac{|\phi_m(E)|}{\#\ moving\ cards}}{\#lists}$

References

1. Peña, J., Bravo, A., del Río-Ortega, A., Resinas, M., Ruiz-Cortés, A.: Design Patterns for Board-Based Collaborative Work Management Tools. In: Int. Conf. on Advanced Information Systems Engineering. pp. 177–192 (2021)