

⚠ Sie haben keinen Zugang mehr zu unserem Produktsupport und Updateservice. Online verlängern, später erinnern oder nicht mehr erinnern.

Seiten / ... / MeetingMirror 2014

## MuC2014 UserActivityLogging

Erstellt von Eva Lösch, zuletzt geändert von Mandy Balthasar am Sep 25, 2018

⚠ Achtung: Diese Seite wurde auf den neuen Wiki-Server im RZ umgezogen - unter <https://wiki.unibw.de/display/MCI/> - Bitte hier keine neuen Inhalte mehr anlegen und auch keine Inhalte mehr editieren.

### logfile-3.xlsx

## User Activity Logging for MuC2014

### Beispiel Logfile: logfile-3.xlsx

Logging							
Message	Particle zoomed to next visual state!	Particle zoomed to previous visual state!	Particle dragged!	Particle rotated!	Search Component opened!	Search Component dragged!	Search operation perform
UserAction	particleToNextVisualState	particleToPreviousVisualState	draggedParticle	rotatedParticle	openedSearchComponent	draggedSearchComponent	✗
GestureEvent	touchTapped/zoomZoomed	zoomZoomed	dragFinished	rotationFinished	touchHold	dragFinished	✗
ConsequenceAction	✗	✗	✗	✗	✗	✗	searchPerformed
TimeStamp	✓	✓	✓	✓	✓	✓	✓
VisualComponentID	✓ <i>particleID</i>	✓ <i>particleID</i>	✓ <i>particleID</i>	✓ <i>particleID</i>	✓ <i>searchingVisualGroupID</i>	✓ <i>searchingVisualGroupID</i>	✓ <i>searchingVisualGroupID</i>
DataType	✓	✓	✓	✓	✗	✗	✗
Title	✓	✓	✓	✓	✗	✗	✗
FromVisualState	✓	✓	✓	✓	✗	✗	✗
ToVisualState	✓	✓	✗	✗	✗	✗	✗
NumberOfImages	✓	✓	✓	✓	✗	✗	✗
NumberOfTags	✓	✓	✓	✓	✗	✗	✗
CurrentPositionX	✓ <i>particlePositionOnScreen</i>	✓ <i>particlePositionOnScreen</i>	✓ <i>particlePositionOnScreen</i>	✓ <i>particlePositionOnScreen</i>	✓ <i>searchingVisualGroupPositionOnScreen</i>	✓ <i>searchingVisualGroupPositionOnScreen</i>	✓ <i>searchingVisualGroupPx</i>
CurrentPositionY	✓ <i>particlePositionOnScreen</i>	✓ <i>particlePositionOnScreen</i>	✓ <i>particlePositionOnScreen</i>	✓ <i>particlePositionOnScreen</i>	✓ <i>searchingVisualGroupPositionOnScreen</i>	✓ <i>searchingVisualGroupPositionOnScreen</i>	✓ <i>searchingVisualGroupPx</i>
ZoomFactor	✗/✓	✓	✗	✗	✗	✗	✗
DragStartPositionX	✗	✗	✓	✗	✗	✓	✗
DragStartPositionY	✗	✗	✓	✗	✗	✓	✗
DragEndPositionX	✗	✗	✓	✗	✗	✓	✗

<b>Logging</b>							
DragEndPositionY	✗	✗	✔	✗	✗	✔	✗
TouchTappedPositionX	✔✗	✗	✗	✗	✔	✗	✗
TouchTappedPositionY	✔✗	✗	✗	✗	✔	✗	✗
RotationAngle	✗	✗	✗	✔ <i>is always 0</i>	✗	✗	✗
RotationPositionX	✗	✗	✗	✔	✗	✗	✗
RotationPositionY	✗	✗	✗	✔	✗	✗	✗
ParentGroupID	✔ <i>flowItems or graph</i>	✔ <i>flowItems or graph</i>	✔ <i>flowItems or graph</i>	✔ <i>flowItems or graph</i>	✔ <i>flowView</i>	✔ <i>flowView</i>	✔ <i>flowView</i>
ParentGroupClassName	✔ <i>flowItems or graph</i>	✔ <i>flowItems or graph</i>	✔ <i>flowItems or graph</i>	✔ <i>flowItems or graph</i>	✔ <i>flowView</i>	✔ <i>flowView</i>	✔ <i>flowView</i>
SearchTerm	✗	✗	✗	✗	✗	✗	✔
NumberOfSearchResults	✗	✗	✗	✗	✗	✗	✔

Keine Stichwörter

2 Kommentare



Ott, Florian sagt:

Nice, Eva, die Tabelle liefert wie erhofft auf jeden Fall einen deutlich besseren Überblick als die einzelnen Tickets. Hier mal die Dinge, die mir zusätzlich zu den per Skype besprochenen Einzelheiten noch aufgefallen sind:

1. Wenn der Rotationswinkel immer 0 ist, wie ist das Ganze denn in RotationPositionX und RotationPositionY codiert? Hier fände ich es auf jeden Fall besser, zu einem Partikel (vielleicht auch VisualComponent?) generell bei jedem Start einer Aktion und beim Ende einer Aktion jeweils den RotationAngle (ggf. vorberechnet) zu speichern. ), also StartRotationAngle und EndRotationAngle.
2. Würde es nicht Sinn machen, auch bzgl. der Positionen (x,y) nur je EINE StartPositionX, StartPositionY und EndPositionX, EndPositionY (jeweils immer linke obere Ecke oder ggf. auch Mitte des Partikels) zu speichern, dann wäre das einheitlich für alle Events und man könnte die anderen Variablen wie folgt abbilden: StartPositionX/Y = CurrentPositionX/Y = DragStartPositionX/Y = ? RotationPositionX/Y; EndPositionX/Y = DragEndPositionX/Y = ? RotationPositionX/Y. Falls sich die EndPosition nicht ändert, würde man sie entsprechend einfach mit der StartPosition belegen.
3. Das mit der TouchTappedPosition verstehe ich noch nicht ganz. Aus meiner Sicht gibt es nicht zwingend nur eine, sondern ggf. mehrere, z.B. beim ZoomEvent entsprechend zwei. Nachdem wir keine Dreifingergesten haben, würden zwei aber auch ausreichen (Zoom, Rotate, Scroll). Wenn ich das richtig sehe, müsste man das auf jeden Fall zusätzlich loggen, denn die TapStartPositionX/Y|1/2 (wo sind die beiden Finger, ggf. einer null) wäre ungleich der StartPositionX/Y (wo ist die linke obere Ecke bzw. Mitte des Partikels). Gleiches gilt für die EndPosition.
4. Bezogen auf (3) und (2) wäre aus Multitouch-Sicht für die zukünftige Verbesserung der Usability eigentlich besonders interessant, welche UserActivities NICHT ⚠️ erfolgreich sind, also wenn z.B. versucht wird zu zoomen, aber der Zoom außerhalb der Partikelgrenze stattfindet und deshalb nicht als zoomZoomed mit particleToNext/PreviosVisualState interpretiert wird. Das wird aufgrund der aktuell sehr klein zoombaren Partikel vermutlich sehr häufig passieren. Können wir das auch irgendwie loggen?
5. Rein Verständnshalber: Für was brauchen wir eigentlich TriggersCreatingGraph, denn das sollte doch immer dann passieren, wenn ParentGroupID = null, FromVisualState = Preview und ToVisualState = Detail, wäre also "berechenbar", oder?
6. Brauchen wir beim ZoomFactor nicht eigentlich auch einen StartZoomFactor und EndZoomFactor, um einen Zoom-Event abzubilden?
7. Verstehe ich es richtig, dass die "... left screen" über die entsprechende ID den detaillierteren Informationen zugeordnet werden können und diese deshalb nicht mitgeloggt werden (z.B. Title eines Partikels). Das Logging finde ich hier generell noch etwas weit weg vom User. Bräuchte man hier nicht eigentlich besser einen generischen FlickEvent (analog zum DragEvent) mit zusätzlichem Attribut (leftScreen = true / false - dieses Flag dann auch für draggedParticle - vielleicht bräuchte man es ja auch gar nicht, weil uns ja die ScreenBounds jeweils bekannt sind oder man könnte die Koordinaten in diesem Fall auf -1 setzen?). Generell: wie bildest du den Flick - also das Werfen - sonst ab? Dazu ebenfalls passend: Ist visualComponentLeftScreen nicht eigentlich redundant, da ja im Endeffekt alles eine VisualComponent ist? Also würde der Event bei einem Partikel beispielsweise doppelt geloggt, einmal beim particleLeftScreen und dann nochmal durch visualComponentLeftScreen?
8. Was aus meiner Sicht generell noch fehlt sind die besprochenen Events, wenn ein Partikel im Flow auf dem Screen auftaucht (keine UserInteraction) und wenn etwas vom Stack genommen wird (leftStack, rightStack, searchStack - hier inkl. Logging des zugehörigen SearchTerms).
9. Die Timestamps sollten wir direkt in einem in Excel lesbaren Format loggen (s. auch Anmerkung zum CSV unten).
10. Brauchen wir die Message und Gesture überhaupt, oder ist das nicht eigentlich alles redundant und nicht weiter hilfreich für die Auswertung?
11. SearchTerm könnte man vielleicht als Title "verpacken" und NumberOfSearchResults (im Hinblick auf den SuchstackDrag) vielleicht generisch als ElementCount abbilden.
12. Bzgl. Keyboard fände ich es auch spannend, wie oft sich jemand korrigiert, bevor er am SearchTerm "ankommt". Vielleicht sollten wir hier einfach alle Keys mitloggen (auch Backspace, etc.). Das sollte sich ja generisch einbauen lassen.
13. Wie steht's denn mit Tags: nachdem das keine Partikel sind, fehlt das Logging hier noch komplett oder?
14. Wie steht's mit der Uhr?
15. Je mehr Typen hier hinzukommen (Clock, Tags, Search, Key, etc.), desto mehr würde ich - wie schon per Skype - dafür plädieren, den DataType nicht für den MashupDataType, sondern für den VisualComponentType vorzusehen und alle Events generisch zu loggen, also componentOpened (DataType € (Clock, Search, PersonParticle, OrganisationParticle, Tag, etc ...)); analog für componentDragged, componentRotated, componentZoomed (Keyboard / Stack / etc. hätte hier halt nie einen Event), componentFlicked. Wenn man jetzt noch den StartVisualState und den EndVisualState speichern würde (glaube, das machst du ja in FromVisualState und ToVisualState, könnte man auch einen generischen componentTappedEvent nehmen, der parallel zum componentZoomedEvent funktionieren würde. Insgesamt sähe deine Tabelle oben dann anhang einiger Beispiele in etwa so aus (gleich mal für CSV / Excel entsprechend gespiegelt:

Event	Timestamp	ID	ParentID	ParentClass	Type	Title (=SearchTerm)	Finger1X	Finger1Y	Finger2X	Finger2Y	StartX	StartY	EndX	EndY	StartVisualState	En
componentTapped	2014-08-16 22:13	2	-	InfoParticle ?	PersonParticle	Florian Ott	100	200	-	-	70	140	70	140	Micro	Pre
componentZoomed	2014-08-16 22:14	2	4 *1	Graph ?	PersonParticle	Florian Ott	110	190	150	210	70	140	35	70	Preview	De
componentRotated	2014-08-16 22:14	3		VisualComponent ?	Clock	-	150	170	170	193	150	160	180	150	-	-
componentFlicked	2014-08-16 22:14	2	4	Graph	PersonParticle	Florian Ott	140	160	170 *3	150 *3	180	150	-1 *4	-1	Detail	De
tapHold	2014-08-16 22:15	-	-	Stack ?	-	-	500	430	-	-	-	-	-	-	null	Se
componentTapped	2014-08-16 22:16	KEY_O	5 *6	Stack ?	Key	Teilwo	255	230	-	-	100	200	-	-	-	-
componentTapped	2014-08-16 22:17	KEY_SEARCH	5	Stack ?	Search	Teilwort	300	310	-	-	100	200	-	-	-	-
particleFlow	2014-08-16 22:18	2	-	InfoParticle ?	PersonParticle	Florian Ott	-	-	-	-	1920	700	-	-	Preveiw	Pre
componentDragged	2014-08-16 22:19	4	6	leftStack	EventParticle	Usability Track	30	40	-	-	20	35	78	100	Micro	Mik

\*1 wäre dann die GraphID

\*2 hier könnten wir dann jeweils die Anzahl der ChildNodes / RelatedItems abspeichern.

\*3 eigentlich nur ein Finger. Finger2Post würde quasi den DragEndpunkt abbilden, also den Teil der Bewegung, bei welcher der Partikel zuletzt "unter" dem Finger war.

\*4 außerhalb des Bildschirms (= leftScreen).

\*5 hier wäre ja auch interessant, ob der Tap & Hold ausreichend lange ausgeführt wird, um die Search zu öffnen, das ließe sich beispielsweise so "codieren".

\*6 wäre in diesem Fall wohl das zugehörige Keyboard bzw. der SearchStack, um mehrere gleichzeitige Suchen zuordnen zu können. Im Title könnte man dann jeweils das bis dato getippte TEILWORT mitprotokollieren.

Wie schon per Mail geschrieben, ist das allerwichtigste aus meiner Sicht, dass die Logeinträge direkt als in Excel einlesbaren CSV mit den "Spalten" der Zeilen deiner ersten Tabellenspalte (bzw. analog zu meinem Tabellenvorschlag) durch log4j geschrieben werden und nicht als studier Text, sonst ist das Ganze niemals auswertbar. Die Hinweise für die Minimalanpassung des entsprechenden Logging-Formatters hatte dir Peter ja schon per Mail geschickt. Wichtig fände ich auch, für das UserActivityLogging das Printing auf die Konsole auszuschalten, sobald in das CSV-File geschrieben wird, damit hier kein unnötiger Datenmüll entsteht. Im Optimalfall ist so ein Log (vgl. meine Tabelle oben als UserInteraktionsfolge sogar menschenlesbar, enthält also NUR Einträge, die sich echten User- oder Systemaktionen zuordnen lassen.

So, genug der geistigen Ergüsse ...



Ott, Florian sagt:

Wie schon kurz telefonisch besprochen, hier als Vorbereitung für ein kurzes Vor-Ort-Wrap-up morgen mal die wichtigsten Dinge (v1), die mir am angehängten Beispiel-Logfile (<http://wiki.informatik.unibw-muenchen.de/confluence/download/attachments/25100350/logfile-3.xlsx?version=1&modificationDate=1409154608634>) zusätzlich zur generellen Strukturdebatte oben (sinnvollerweise eher v2 😊) aufgefallen sind.

1. Bei den Uhrzeiten fehlen leider die Sekunden (oder gar Millis), um die Chronologie herstellen zu können. In Excel ist das Datenformat leider wie oben schon befürchtet nicht als Datum lesbar, dafür bräuchten wir wohl (z.B. entsprechend <http://stackoverflow.com/questions/804118/best-timestamp-format-for-csv-excel>) ein SimpleDateFormat mit "MM/DD/YYYY HH:MM:SS" oder "yyyy-mm-dd hh:mm:ss,000". Alternativ könntest du den Zeitstempel als Java DateTime (toString) ausgeben und wir würden dann ggf. per <http://www.spotlight-wissen.de/archiv/message/1105382.html> in Excel wandeln (Millis bleiben erhalten). Sollten wir aber auf jeden Fall vorab testen. Excel BTW liegt für den MAC unter `\deployment.cscm.informatik.unibw-muenchen.de\CSCM MAC Install (Anmeldung mit CSCM-Domäne)`.
2. Die Koordinaten scheinen mir in irgendeinem relativ abstrusen virtuellen JavaFX-Koordinatensystem errechnet zu sein. Was wir bräuchten (für alle X,Y) sind echte ganzzahlige Werte in unserem FullHD 1920x1080 Raum. Nur so könnten wir z.B. ein post mal so etwas wie eine ScreenHeatmap (wo Partikel vorrangig geöffnet oder auch gedragged / geschlossen werden) erstellen.
3. DragEndPosition ist immer 0.0, sollte vermutlich (zumindest bei draggedParticle) die Endposition beim dragFinished sein. Was ist denn in dem Kontext die CurrentPosition? - Ist das der Tap-Punkt und DragStartPosition sowie DragEndPosition beziehen sich dann auf die linke obere Ecke des Partikels?
4. Wenn ich dich richtig verstanden habe, sollte die ParentGroupID ja beim Graph quasi "der Graph" sein, also insbesondere die Zuordnung der Navigation von Partikel zu Partikel innerhalb eines Graphs ermöglichen. Die ParentGroupID müsste also jeweils anders sein, sobald jemand einen "neuen" Graph aufmacht, oder? Zumindest für ParentGroupID "4" (VisualGroup) scheint das aber aktuell nicht so zu sein, denn hier gibt es die 4 jeweils als ParentGroupID für Zeile 2,4,5,etc. und dann nach dem "destroy" in Zeile 40 wieder ab Zeile 51. Ist das so gedacht? Hast du da einen Graph per Pinch "geschlossen" und dann wieder geöffnet und er behält deshalb quasi den existierenden Graph inkl. ID? Wie bekommen wir damit raus, welche Wege innerhalb eines Graphs "abnavigiert" wurden?
5. RotationPosition und -Angle scheinen ebenfalls immer 0.0. zu sein.
6. Beim DataType fehlt beim Content die Unterscheidung, ob es ein Event, Bild, etc. ist, kann das sein? Zumindest sieht ein P1110326 sehr nach einem Bild aus, hat aber nur `org.sociotech.communitymashup.data.impl.ContentImpl` (nicht die abgeleitete tatsächliche Klasse).
7. Den Unterschied zwischen TagImpl und MetaTagImpl (aus Nutzersicht) verstehe ich noch nicht ganz; so oder so fehlt im Title dazu der entsprechende Tagname (ist aktuell ja leer).
8. Ist die VisualComponentID = ID des Dataltems im Mashup oder wie stellen wir hier sonst den Bezug her?
9. Wie kann eine Person "Ulf Schubert" (erster Datensatz) 6 als NumberOfImages haben? Hat der Mann wirklich sechs Profilbilder? Wo kommen die denn her?
10. Wie lässt sich denn die Suche in Zeile 54 einer der beiden Search Komponente (aus 49 / 52) zuordnen?
11. Wo sieht man, ob etwas aus (welchem) Stack gedragged wurde? Hier hätte ich beispielsweise erwartet, dass z.B. in der ParentGroupName zumindest ein "Stack" auftaucht ... denn über die Koordinate des DragStart könnte man ja ableiten, ob links oder rechts. Für den Suchstack wäre es aber sehr wichtig, zusätzlich der SearchTerm zu "merken", also ob ein Micro "Florian Ott" aus dem SearchStack nach SearchTerm "ott" gedragged wurde.
12. Was mir auf jeden Fall zu fehlen scheint und was sehr wichtig wäre, sind Flicks aus dem Bildschirm (schließen) - zumindest habe ich das oben beschriebene particleLeftScreen nicht bei einer der SearchComponents oder einem Graph und auch nur bei einem Partikel finden können (hast du das vielleicht einfach nicht gemacht?)
13. Ansonsten fehlt das Logging des Flows - also was wann auf dem Bildschirm erschien und wann wieder verschwand.
14. Grundsätzlich wäre sicher noch ein AutoClose-Logging sinnvoll, damit wir mitkriegen, wenn etwas per Timer "geschlossen" wurde (ansonsten sucht man sich bei der Auswertung leicht tot, wo etwas hingekommen ist). Sofern man sich das herleiten kann oder es zu viel Aufwand macht, aber besser v2.