

Joseph Kubiniec



Copy of Dashboard Experience Routing



Owned by Joseph Kubiniec ...

Apr 03, 2023 · 4 min read · 4 people viewed

Problem Statements

- As a **non-technical experience manager**, the current implementation of experience routing is too complex (Building gatekeepers). It requires fluency around macros and programming logic
- As an **experience manager**, the current implementation of experience routing requires managing too many "gatekeeper" experiences, causing me confusion and lost time as I manage gatekeeper associations with end destination experiences

Experience creators need a more intuitive, less technical, and more succinct way to design and maintain consumer journeys across experience files.

UX Research

Scope and roadmap planning appear essential in defining this story.

Problem

Users need to be able to assign any Experience destination to any touchpoint – regardless of whether the touchpoint's associated object are in a project.

Jason S rephrased it like this:

"users should be able to assign a touchpoint to an experience at any point along its child-parent line of inheritance: touchpoint->experience ;

touchpoint->object->experience ; touchpoint->object->project->experience "

Current system state

1. Objects are currently required to be contained with a Project
 - a. This will be changed in the post Blue Bite Lite world
2. A Project is currently limited to a single Experience destination for all of the objects contained
3. Objects can be assigned multiple tpoints yet all of the touchpoints go to the same Experience destination determined by their associated Object

In effect, in today's world, every Touchpoint, through it's associated Object within a project has to direct to the exact same Experience destination. It is only at Experience level can any kind of routing logic be applied,.

Why Is This A Problem

The inability for users to assign different routing destinations to Objects (and Touchpoints) within a project presents difficulty in creating and managing complex consumer journeys.

Currently users have to expend resources crafting workaround solutions resulting in unnecessarily bloated Studio files to accomplish **some** of what they want.

The soon to be history limitation wherein Objects need to be contained within a Project (gone with the work done by BB Lite) essentially makes the Object – and possible the Touchpoint – a front and center element of the user journey experience creation. It therefore needs far more functionality in customizing the users' experience as well as ease the process flow of the Studio developer.

Hypothesis for Solution

Blue Bite will provide a set of routing value/pairs in which logic can be applied - **Experience Routing** - to determine the specific Experience/URL that the

specifically activated Touchpoint is routed to. Simply for example, a few routing value/pairs:

- Location Based Routing; ex: "Finland Go To Experience A"
- Language Based Routing; ex: "English Go to Experience B"
- Device Based Routing; ex: "Android: Go To Experience C"

ISSUE: Is "sequencing" (stringing) the Experience Routing options to be considered? The output of one acts an "input" into another? Logic "within" the specific routing types but also then, "logic" between the types themselves?

MORE PLAINLY: In the scenario above, where would a user in Finland who speaks English and is using an Android phone be directed?

As Experience Routing logic can be applied at the Project, Object and Touchpoint levels a **precedence logic** needs to be defined for determining flow. Below is a breakdown of a possible logic tree. A flow chart below also provides this flow.

Assumption: The act of beginning the flow is the act of a user reacting to a code. Flow of logic from point of scan:

Touchpoint

1. Has No Routing Logic
 - Refers to associated Object for instruction
 - Default/"Whoops" needed if Object connection error
2. Has Routing Logic
 - Overrides any instruction from associated Object
 - Includes rules for "whoops" experience

Object

1. Has No Routing Logic
 - If associated with a Project, Project provides instruction
 - If not associated with a Project a default destination must be provided
 - Default/"Whoops" needed if Project connection error
2. Has Routing Logic

- Overrides any instruction from associated Project
- Includes rules for "whoops" experience

Project

If Object being called contains no-overriding experience routing:

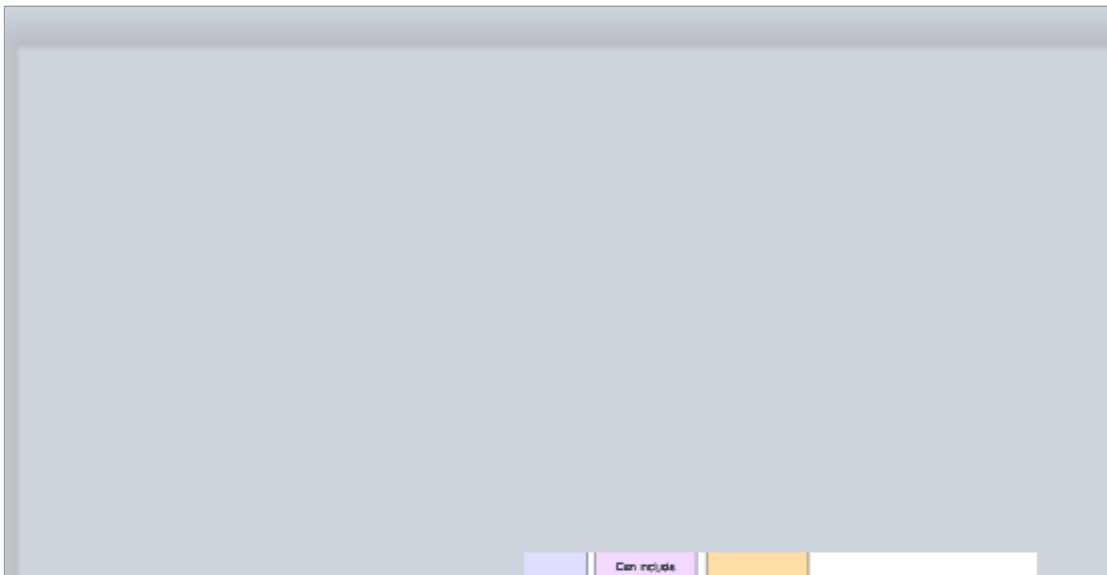
1. Project Has No Routing Logic
 - Object is directed to destination instructed by Project
 - Default/"Whoops" needed
2. Project Has Routing Logic
 - Object is assigned the destination determined by Projects Experience Routing logic
 - Default/"Whoops" needed

Experience Destination

- The determined Experience Destination is then presented to the user
 - Experiences can contain internal routing logic with Studio files
 - This is not part of scope of work considered here.

Flow diagrams illustrating a proposal for the hierarchical flow of permissions.

EXAMPLE Of Destination Flow. Did i sat "This Is a Comp For Discussion"? 😊



Solution Hypothesis Requirements

MOST CRITICAL FOR SUCCESS: UX and Logic must be delivered at a high level to decrease the users dependence on techno complexities to get in the way of solving this problem. To solve the problem of enabling user create automated Experience Routing; some requiremets:

1. Projects need to support more than a single routing destination (Experience Routing) for its objects
 - a. Projects can have multiple Objects; each with a possible unique destination.
 - b. Any Touchpoint or Object that does not have Experience Overrides and is contained within a Proiect will have their destination determined by the Project
2. Each Objects need to support more than a single destination (Experience Routing) for itself.
 - a. Many Touchpoints - each with a possible unique destination - can be associated with a single Object
 - i. Object Overrides take precedence over Project rules

3. Each Touchpoint needs to support more than a single destination (Experience Routing) for itself.
 - a. There can be many Touchpoints associated to an Object. They must each be able to define custom Experience Routing
 - i. Touchpoint overrides take precedence over both Object and Project rules
 - b. Any Touchpoint that does not have Experience Overrides will have their destination determined by the Touchpoints' associated Object.

User Benefits of Solving Problem

- Reduce the size of Experience files
- Reduce user errors in Experience routing (whoops landings) due to over-complicated logic
- Enable the creation of my complex and engaging consumer journeys

Primary Examples of Experience Routing For Automation/Logic

- Language
- Location
- Link Type
- Device
- Day Cycle
- # of scans

Business/Roadmap Questions

Decathlon is a big client. They have needs to define routing logic at the touchpoint level.

The majority of our remaining clients would be satisfied with Project level experience routing automation.

Ideally the UX solution for Experience Routing at the touchpoint level would be the same or similar than doing the same at the project level.

How do we prioritize delivering if we have to focus our initial release on one or the other use cases?

Primary Persona(s)

- Experience manager

Use Cases

1. As a user i want to create logic that can be associated with a project that – depending upon variables incoming from the the activated touchpoint – that can direct the user to the landing page destination that matches the language settings of the users device.

+ Add label



Be the first to add a reaction

