

Technical Disclosure Commons

Defensive Publications Series

November 17, 2016

Differentiated and Personalized Customer Support

Vanessa Harris

Melinda Porebski

Michael Berg

Sneha Chaturvedi

Nicholas Croll

See next page for additional authors

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Harris, Vanessa; Porebski, Melinda; Berg, Michael; Chaturvedi, Sneha; Croll, Nicholas; Katragadda, Geetika; Feld, Brandon; Redda, Merhawi; Lu, David; Tucker, Joshua; and Chen, Ping, "Differentiated and Personalized Customer Support", Technical Disclosure Commons, (November 17, 2016)

http://www.tdcommons.org/dpubs_series/318



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Inventor(s)

Vanessa Harris, Melinda Porebski, Michael Berg, Sneha Chaturvedi, Nicholas Croll, Geetika Katragadda, Brandon Feld, Merhawi Redda, David Lu, Joshua Tucker, and Ping Chen

Differentiated and Personalized Customer Support

BACKGROUND

Customer support systems can provide support to customers in various ways. For example, support agents can solve support problems for customers who have problems with applications or with systems or products available over the Internet. Effective customer support systems can improve the outcome of support, increase revenue, and lead to higher customer retention and loyalty.

SUMMARY

An entity, such as a support agent, can provide customized service (e.g., treatment) to a customer based on the cohort(s) of which the customer is a member. The treatment that a customer receives can be based on treatments associated with the group, or the customer can receive personalized treatment if detailed personalization information based on customer-specific attributes has been received and stored.

DESCRIPTION OF DRAWINGS

Figure 1 shows example operations for assigning cohorts and obtaining treatments for the cohorts.

Figure 2 shows example operations for collecting and reporting metadata for activities.

Figure 3 shows example operations for displaying cohort information.

DETAILED DESCRIPTION

In order to provide differentiated and personalized support, “cohorts” (e.g., customer groups) are created that are used to group customers that have a common set of attributes. For example, a cohort can exist for a customer group that includes customers that spend a certain amount on advertising, and a different cohort can include customers that have access to a particular feature, product or service. The common set of attributes for customers in a cohort can be determined from a variety of signals, such as through product use, the receipt of marketing emails, previous support interactions, or other signals that can be used to group customers based on similarities. Each cohort can have an associated canonical name or other such identifier that uniquely identifies the cohort, such as in the form <Product>.<Signal>.<SignalValue>. Signals can indicate cohort eligibility for each customer. Each cohort can also have a cohort display name, such as displayed to support agents providing support, that is a plain English and support user-friendly name for the cohort. In some cases, cohorts can be created based on a collection of signals and a set of associated business logic that is applied to them. For example, business logic and the collection of signals can be used to generate a single signal that represents a cohort.

Once cohorts have been created, treatments can be defined for each cohort. Treatments can include a set of actions that a support agent or an automated process, for example, should take for (or with) a customer. Defining treatments, for example, can improve support models and optimize support for ad campaigns, thus improving the customer’s business. Each cohort can be configured to display, e.g., in a user interface, appropriate treatments that describe how support agents are to operate (e.g., interact) with a customer who belongs to the cohort. When a customer belongs to more than one cohort, cohorts and their treatments can be prioritized by importance. In some cases, actions from multiple cohorts can be blended into a

single list, including removing any conflicting and/or duplicated actions across cohorts. A treatment family can be a set of treatments for customers in a cohort. Information about cohorts and treatments can be stored, for example, in a knowledge management system that can facilitate access by low latency application programming interface (APIs) that fetch stored data as needed, such as during a customer support call. Cohorts can also be used to drive publically-accessible knowledge base articles, and to provide unique personalized support directly to customers based on the same set of signals about a customer. Further, cohorts can also be used to drive experiments, such grouping agents in experiments that use support models. In some cases, users can define treatments that are to be taken, as part of the cohort, on behalf of the user.

Cohorts can be defined and configured as a long-running program, or specifically for control groups and experiment groups. For example, an experiment involving cohorts and treatments can be incorporated as a more permanent program if the experiment is successful. Platforms created for programs and experiments can provide operational teams with the flexibility to personalize treatments for cohorts and to experiment with treatments that drive business metrics. The experiments also allow for tighter integration between support, marketing and sales efforts. Information from experiments can include customer interaction histories that can be used by support agents in support actions. In an experiment/program, a control group can be defined to include a subset of customers (of a cohort) who are not given the experiment treatment. Control groups can be used so that comparisons can be run (e.g., control group vs. experimental group), such as to determine the effects of treatments.

In some situations, detailed personalization information based on customer-specific attributes can be captured, stored and used. For example, a cohort can be defined for users who are running out of space within a mail application. Most users may be given the option to

upgrade their accounts, e.g., to obtain additional space. However, based on how long a user has been a customer (or how much the user has spent, such as in an application store), some users can be offered a discount or a free upgrade in lieu of being granted additional space. In this way, the cohort can be used to extend a basic system behavior to personalize support by user while still preserving an intent for the group (e.g., by solving a mail storage space problem).

Figure 1 shows example operations for assigning cohorts and obtaining treatments for the cohorts. For example, a user using a front end can assign cohorts based on topics related to users for use in defining respective cohorts. Cohorts can also be assigned using algorithms, for example, that automatically assign cohorts based on information about users and topics.

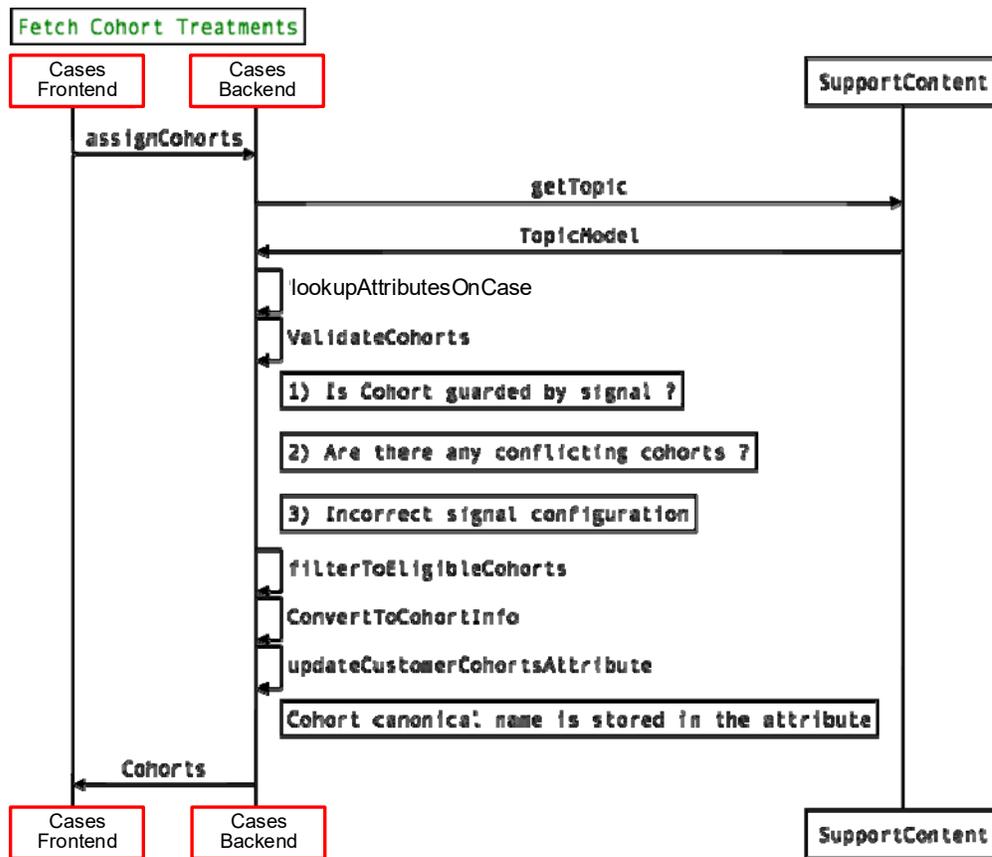


Figure 1

A request can be provided by the frontend to a backend that performs most of the operations. For example, the backend can access support content in order to obtain a topic model associated with a given cohort. The backend can look up attributes associated with the cohort. The backend can perform validation on the information including, for example, determining if the cohort is guarded by a signal, identifying any conflicting cohorts, and determining if incorrect signal configurations exist. The information can be filtered and converted into cohort information form. Customer cohort attributes can be updated to identify the new cohort, and a cohort canonical name can be created and stored. The created cohort information can be provided by the backend back to the frontend, e.g., for presentation to the user.

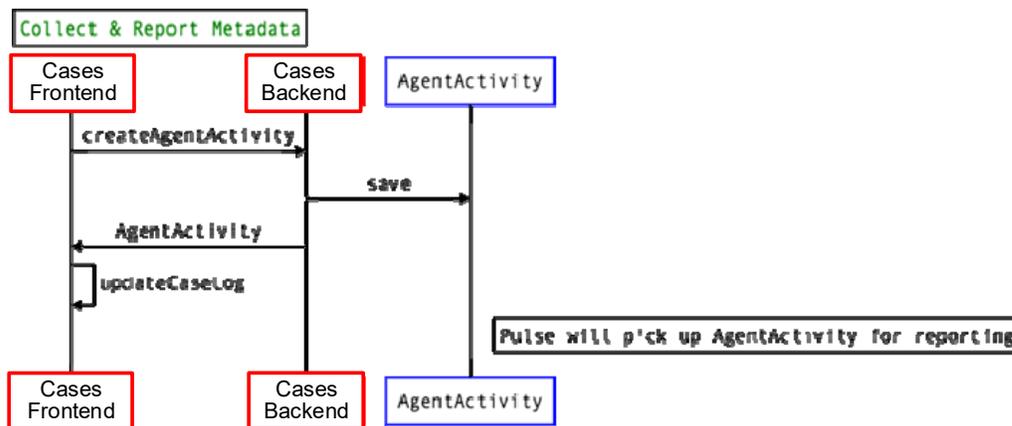


Figure 2

Figure 2 shows example operations for collecting and reporting metadata for activities. For example, at the frontend, a request can be made to create agent activity. The request can be processed by the backend, which can initiate an agent activity and enable reporting. The backend can indicate to the frontend that the request has been handled, and the frontend can update its case log.

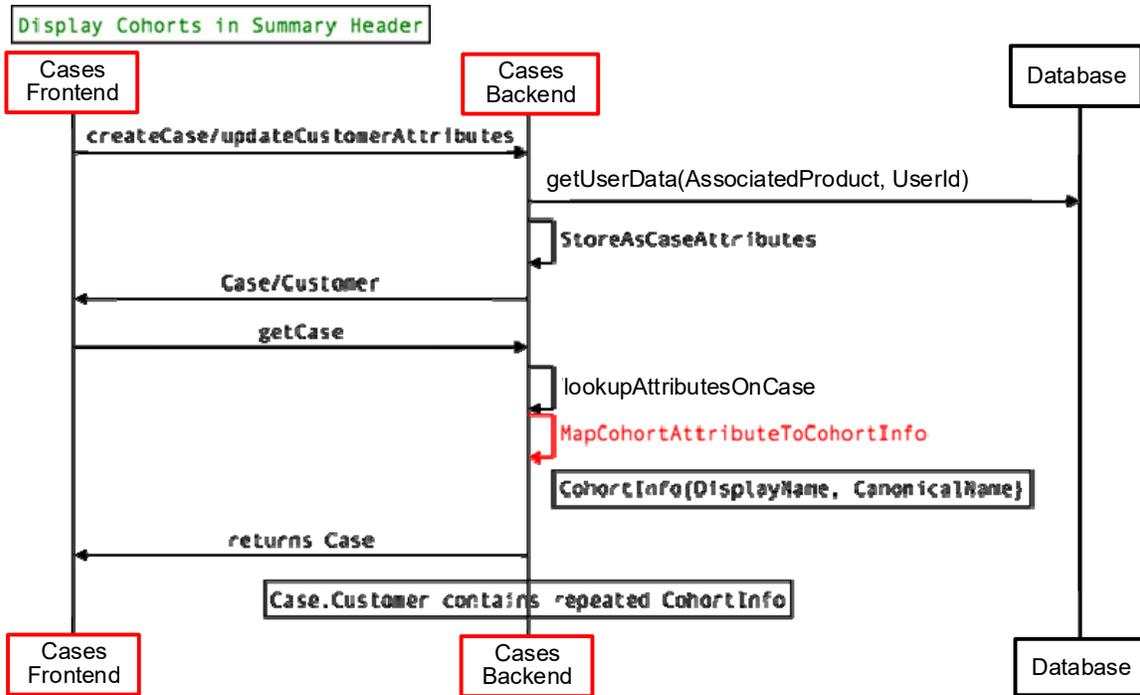


Figure 3

Figure 3 shows example operations for displaying cohort information. The frontend can provide a request to the backend to create a case and update customer attributes. The backend can access the database to get user data associated with a product and a user ID, and the backend can store case associated attributes. For the case identified by the front end, the backend can look up corresponding attributes, map the attributes to a cohort, and return the cohort information to the front end for presentation to the user.

ABSTRACT OF THE DISCLOSURE

An agent can provide customized service (e.g., treatment) to a customer based on the cohort(s) of which the customer is a member. The treatment that a customer receives can be based on treatments associated with the group, or the customer can receive personalized treatment if detailed personalization information based on customer-specific attributes of the customer has been received and stored.