Self-Optimizing User Interface

This invention automatically revises and customizes a user interface to maximize or minimize performance against an objective.

Based upon a set of input conditions and an objective function, the invention adapts the user interface that will be presented to a specific user in real time or near real time. Examples of objective functions include:

- Maximize the number of times user's click on the buy button
- Minimize the number of user actions needed to find and play an on-demand video
- Minimize the number of bytes that need to be transferred over the network in order to get the user what they probably need
- Maximize the number of advertising spots or display ads shown to users.
- Maximize the number of choices that are quickly presented that are geographically close to the user's location

Input conditions can include

- The identity of the user engaging the user interface right now.
- Historical record of user actions and timestamps in the user interface (clicks, keystrokes, taps, swipes, voice commands, gestures), either for the particular user using the interface right now, or for a broader population of users.
- User preferences
- The device currently in use,
- The network in use, and current or expected network performance
- Time of day and day of week
- Geographic location or proximity to a specified location

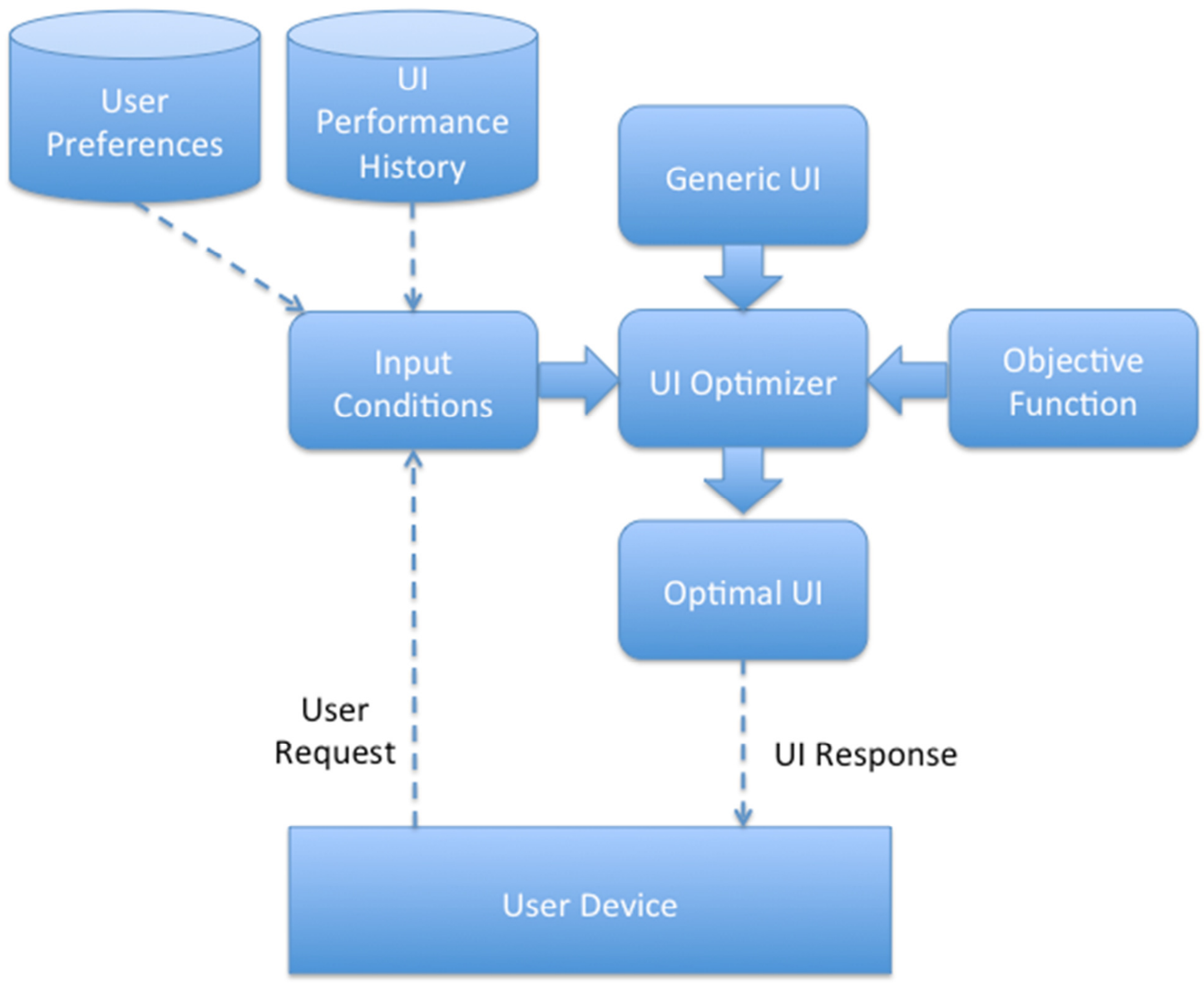

Figure 1: High-Level Diagram of a self-optimizing user interface.

The UI Performance History is a historical record of the way users used the interface in the past, potentially including the particular user using the interface right now. The data about how most users "clicked through" a set of options on a series of web pages to reach a specific goal can be used to optimize the "menu tree" that the user has to navigate, so that the user can more easily reach the desired objective. The invention may add to a UI Performance History databases used to manually re-design web sites to improve their performance against a set of objectives. In this invention, the UI Performance History is used to automatically re-create the user interface to achieve a specific objective based upon the context of the user request.

In one example implementation, the User Device is running a web browser and the user causes the device to send a HyperText Transfer Protocol (HTTP) message to get a web page. The generic user interface is defined in one of the versions of the HyperText Markup

Language (HTML), The UI Optimizer reads the Generic UI HTML, determines that given the objective function, it would be best to restructure a menu tree so that some options appear on the first web page and others on later pages, and re-writes the HTML in real time to create the Optimal UI, and then sends the Optimal UI to the User Device in an HTTP message.

In another example, the changeable parts of the user interface can be written or re-written in an interpreted language such as Java.

The UI Optimizer is a key part of this invention. It can be implemented in a number of different ways, including the following examples:

- Leveraging an automated statistical analysis of historical use of the interface to determine which user choices are most common for the population at large, for users similar to the current user, or for the current user themself, and using that knowledge to adjust the order that options are presented to the user in order to reduce the number of user actions, or the time required, in order to provide the user what they want.
- Leveraging optimization techniques from the operations research field, including network optimization algorithms, linear programming techniques, and integer programming techniques to predict the optimal user interface for a given user, for a population group, or for the population at large, based on a specified objective function.
- Leveraging an analysis of the byte load generated by specific parts of the user interface so that when the objective function is to minimize the number of bytes delivered to the user, interface elements that require more bytes may be de-prioritized or replaced with other options that convey the same information with fewer bytes.
- Leveraging the knowledge of the device being used to optimize the performance of a particular web site based on the device capabilities, so that devices with low computing power receive fewer interface elements that require local computing and instead receive elements that are pre-computed by the web site itself or by ancillary processors.

The invention can be implemented so that is automatically adjusts user interfaces in real time or in near-real time. Real time automation requires that the optimal UI be created very quickly, so that the user is likely to be unaware of a delay in the response to their web site request. Near-real time implementations of the invention could allow for some categorization of users into groups, and pre-determination of the user interface to present to each group of users based upon an automated process that runs periodically.

User interfaces that recommend particular titles or episodes for viewing in a video on demand application, typically in combination with recommendation engine algorithms, are a subset of this invention. This invention can use the optimization techniques described here to provide a recommendation for a specific user or group of users. However, this invention is by no means restricted to recommendation engine applications.

The invention contemplates a UI Performance History database is used to examine the way that users navigate a user interface, and that information is used to manually re-design the way users flow through a web site user interface. Such a system may not facilitate user interfaces are partially or completely re-written in real time or near real time in response to a user request based upon current context.

The outcomes enabled by this invention may include recommendation engines and location based user interfaces and/or a system for dynamically re-creating a user interface in an automated way based on context and an optimization goal is claimed.