SYSTEM AND METHOD FOR BEHAVIOURAL AND CONTEXTUAL DATA ANALYTICS

FIELD OF THE INVENTION

The present invention generally relates to wireless devices and communications networks. In particular the invention concerns centralized processing and interpretation of behavioural, contextual and optionally technical observations conducted by wireless devices.

BACKGROUND OF THE INVENTION

Solutions for analyzing people's behaviour and usage habits via TV and radio terminals have already existed for some time and more recently, various solutions to measure online usage in the Internet have been introduced. The medium with which many people spend, however, increasing face time is actually a wireless device, such as a mobile phone. Newer phones, so-called smartphones, are increasingly capable of performing other tasks than mere communication functions. For example, it is possible surf on the web with these devices, consume multimedia content (such as music), play games, and access maps, not to talk about the variety of communication services available. In brief, lots of different kinds of usage, in a unique contextual and technical environment, takes place in smartphones, the global penetration of which is constantly increasing, and the importance as a medium being correspondingly elevated.

It therefore seems that in the light of rather dominant near-future trend, various smartphones might emerge as the sole digital devices that people really bother to carry with them on a daily basis. They will function as communication devices, authentication tools, watches, digital wallets and keys, etc. all alike. In this scenario, smartphones could also be exploited as always-on observers of life. Smartphone devices are possibly in the best position to become universal digital (life) observers, being able to track locations, temperatures, movements, communication activities, proximity to other people, social interactions, etc. There is no strict limit with regard to the possible observations as different kinds of observer entities and sensors may be embedded in or connected to the smartphones. Contextual environment is therefore particularly strong in the usage of smartphones. For example, in the future the smartphones might collect a rich feed of data containing

comprehensive audio and video recordings relative to each day, and thus enable the users to generate a digital storage of their life lived. Even today's smartphone applications may be configured to track context, such as location and time, through publicly available application programming interfaces. Data that can be collected in a smartphone may naturally be used locally in the same device. For instance, obtained temperature data can be shown on the screen of the phone. However, some data could be, at least occasionally, worth sending to remote entities.

A number of prior art arrangements specifically propose exploiting information on the usage of data in wireless devices for providing personalized service in return. For example, publication US2004181604 discloses a method for enhancing the relevance of content provided to communication devices, comprising: pushing content notifications relating to a plurality of topics to a communication device via a notification service; receiving content usage information from the communication device; modifying the topics targeted for the communication device based on the content usage information of the communication device; and pushing the content notifications related to the modified topics to the communication device via the notification service.

Although various prior art solutions mention collecting data from wireless devices, they do not really provide tangible means for proceeding with data analysis, exploration and exploitation. The solutions simply disclose collecting some predetermined data items from wireless devices or executing generic analytics regarding the usage of e.g. the Internet with simple statistical tools, for example. Moreover, the contemporary techniques highly rely on manual labour, are more or less use-case centric, typically monitor only a very limited number of events according to fixed and substantially memoryless schemes, and store and distribute the gathered data basically as is with narrow analytical scope.

SUMMARY OF THE INVENTION

Embodiments of the present invention seek to alleviate at least some of the drawbacks of the prior art solutions and provide an intelligent, flexible, effective and adaptive centralized alternative for monitoring and analyzing the usage and context of wireless devices. According to an embodiment, there is provided an automated data mining engine and related platform, which may be implemented as a server arrangement, e.g. software architecture therein. The devised server arrangement may be configured to execute processing and interpretation of behavioural,

contextual, and preferably also technical observations provided by a plurality of wireless devices, operations of an automated data mining engine, and distribution of the derived information through application programming interfaces (APIs) towards external systems. The server arrangement may be applied in various kinds of analysis and/or control applications of both today and the future, which often have significant practical relevance.

According to an aspect, there is provided a server arrangement to process observation data of wireless devices, comprising: data input logic to obtain observation data from the wireless devices the observation data comprising application usage data; data mining logic to process the observation data to identify temporally adjacent applications to generate usage metric data; and data distribution logic to transmit advertisements to the wireless devices based on the usage metric data.

Accordingly, in an embodiment of the present invention, a server arrangement for managing observation data of wireless devices, such as smartphones, comprises

a data input logic for obtaining observation data from a number of wireless devices and optionally supplementary data from a number of other external data sources, the obtained data including behavioural, contextual, and optionally technical raw data relative to the wireless devices,

a data mining logic for establishing a number of derived data elements, such as data tables or databases, on the basis of processing and analysing the obtained observation and optional supplementary data, said processing and analysing incorporating aggregation and optionally correlation, clustering, and/or factoring procedures, wherein at least one derived data element includes usage metrics with contextual and optionally technical dimension relative to one or more applications or other features of one or more wireless devices and users thereof,

a data storage for storing the obtained data and the number of derived information elements, and

a data distribution logic for providing derived data to one or more other entities.

In one embodiment, the data distribution logic may, for example, push data to another entity or serve a data query constructed by the another entity through provision of derived information from one or more derived data elements according to the push logic settings or query parameters, respectively.

5

The contextual dimension may refer to contextual elements, e.g. time or location, in relation to which the behavioural usage data is established. Technical dimension may correspondingly refer to technical data.

Thus, various embodiments of the server arrangement of the present invention preferably exploit multi-dimensional, holistic data stream that may be input thereto substantially continuously from one or more wireless devices. This data stream may include behavioural (for example, start-up timestamp of a particular application session, and/or key press actions via the user interface of the device), technical (for example, bearer of a data session or signal strength during a voice call, data throughput, crash log), and/or contextual (for example, time data, current battery level or current location such as currently active cell of the network) data items. These different kinds of data items may be included in the same data transmission or a plurality of transmissions. Accordingly, the server arrangement is preferably configured to process the incoming behavioural, contextual and/or technical data (for example, to perform data filtering or clustering), enrich it (for example, to add metadata on application names and categorization), analyze it (for instance, execute recognition of behavioural and contextual patterns), build collective dynamic intelligence (for example to receive automatic admin alarms when a new application has been adopted by people, in other words significant behavioural deviations from the past take place) and/or provide derived information to external systems preferably through an application programming interface (API), which may implement a push functionality and/or be queried either on regular intervals or dynamically based on separate triggers. Further, the server arrangement preferably facilitates using (new) applications exploiting the derived information, such as behavioural and contextual advertising applications, and may be easily integrated with them. In addition to programmable intelligence, the server arrangement may incorporate machine learning and adaptive heuristics in understanding the changes in the patterns of wireless device usage, technical problems in data processing, missing data points, etc. that could affect the output of the whole analytics platform.

The input data may thus be initially remotely established by the instances of the applied observer logics running in the wireless devices, for which purpose also some commercially available solution may be applied instead of a proprietary one. Typically one data entity on the level of observations (source data), such as a log row on application session start, which is input, refers to a data point of one or more data items (e.g. measured values), where the data point may relate to e.g. one action or event, but also other configurations are possible. For example, an activation of a device application, with time stamp and identification of the application (ID such as name and/or software identification code), may represent one data row (for instance, behavioural information), a triggered measurement of voice call signal-to-noise-ratio might form another row (for instance, technical information), and the change in the

active cell tower of the network might induce a further row (for instance, contextual information). In the first example, the log row upon application session start contains one data point (session start), which may consists of e.g. two or more data items (name of the application, its application identification code, time stamp, and/or date stamp).

The data flow, or data feed, will be input to the server arrangement through a data input interface thereof. In addition to data input, the data input interface may preferably intelligently cultivate, match, and/or combine data, thus optimizing the use of the raw-level database. One example is the aforesaid enrichment of data items; e.g. the items relating to observed application usage sessions may be supplied with location information such as current cell tower identification code (observed earlier, for instance), which can be later used in matching usage statistics with geo-coordinates, for example. The utilized logic may be straightforward and scalable, whereupon data processing in the data input interface is more contextual than behavioural. In other words, instead of immediately performing, upon data reception, advanced analytics or pattern recognition, the data input module combines, for example, data input log rows that are e.g. consequent to each other according to predetermined criterion, e.g. temporal distance of actions, and/or adds status information such as current cell tower indices (which are stored in status variables, being available all the time) thereto. For instance, in the aforementioned first example the data processing logic makes it possible to combine two consequent sessions of a calendar application together, if there was only a predetermined, e.g. one second, distance between separate sessions (most likely the user just closed a key lock application that automatically started in the user interface after first activating calendar, and this key lock application session was "removed" in filtering the incoming application logs, for example).

Data processing already in the input interface is beneficial as it can be typically done in real-time, reducing the further micro-level data processing load (from centralized databases). By doing certain actions, like combining consequent rows together (if they represent the same session, for instance), the reliability and value of the data can be increased. Further, the data input module may calculate immediate dynamic statistics out of the raw data. Dynamic statistics may provide valuable pieces of information for administrational purposes, possibly not utilizing historical or time series data at all (in contrast, in derived statistics almost everything may be based on aggregation, normalization and/or averaging, effectively processing many log rows in any particular process). Regarding dynamic

statistics, for example, when incoming data is received from a certain user, a counter can be updated indicating that there is some new data from this particular user during a particular time period such as a day, and the time stamps for last data received can be updated for him. With these dynamic statistics it may be easier to follow how the system generally works, and what is the status of data collection.

Considering different embodiments of data storage in the server arrangement, raw data may be stored in a raw level data repository, which may form a part of larger data storage. The data is advantageously stored in relational database(s), and the raw data is frequently provided to a data mining engine for analysis and advanced processing. In addition to storing raw data, the data storage also stores the resulting derived information, i.e. data produced by a data mining module on the basis of conducted analysis. This derived information is preferably in a format that can be directly used by external systems. For example, a logical entity called a data warehouse may be used for storing the directly usable derived information comprising statistics like application usage activity figures for each user or a number of users, and such information can be used in producing aggregated figures such as calculating the penetration of mobile Internet browsers over different device platforms available in the market.

The data warehouse thereby advantageously stores the derived data elements and provides access thereto so that efficient execution of various data queries by the users of the server arrangement is facilitated and quickened further enabling conducting more complex queries and push-mode data provision to external entities. The external entities may include external logical entities physically still integrated with the server arrangement of the present invention, or also physically separate server entities, for example. For instance, queries and data provision on the basis of identifying those wireless device users who are currently abroad and/or have been using the calendar application of the device at least on a weekly basis during the past year, may be performed. Effectively, most data tables or other data entities of the overall data storage are something else than raw data items. Indeed, a number of derivative databases are desirable, and the raw data may be continuously and/or at discrete periods processed by the data mining engine, which then stores the resulting information in the derivative tables or other data elements of the warehouse. The derivative tables may thus incorporate e.g. daily statistics of application usage for each user, whereas in the raw data repository all the information may remain on a session level, unaggregated, for example. On the basis of a received query, derived data may be selected, processed and/or combined for output.

Accordingly, the data mining engine is either discretely or continuously mining the raw data repository and extracting useful information out of the raw data. Further, the data mining engine may mine and/or aggregate already-derived data entities, e.g. derivative data tables, or a combination of raw data and already-derived data entities, for obtaining additional derived data entities. The data mining engine preferably contains customized algorithms and procedures, which may be advantageously updated on the fly, being optimized, (fully) automated, and/or enabled to generate statistics, indicators, barometers, aggregated figures (for example summing up actions), averages (one form of aggregation) and/or other important derived data elements on the basis of the data arriving in the system from the wireless devices and optionally other elements. Further, the data mining engine may conduct factor and cluster analysis, perform correlation calculus, recognize patterns, learn, i.e. adapt its behaviour, from the incoming data, enrich the data and/or in other ways make more intelligence out of the often less meaningful raw data that is collected by the wireless devices, including behavioural, technical and/or contextual data.

A data distribution API may be logically built on top of the rest of the platform in order to efficiently facilitate practical applications and use cases of the platform in view of data provisioning. The API is enabled to serve external clients that request data, i.e. the users of the service arrangement. The request types are standardized, typically including wireless device(s)/wireless device user(s) identification, time period, and/or identification of the needed statistics. In addition to or instead of the pull model of information delivery, the API may push information to one or more interfaces, if needed, by the defined trigger conditions. For example, interaction with a location-based mobile advertising platform may be implemented through the API by a location-based triggering logic.

As to the general utility of the various embodiments of the present invention, they enable conducting automated behavioural, technical, and contextual research including data collection and analytics for various different purposes ranging from targeted behavioural mobile advertising or traffic control (based on e.g. context such as location analysis of wireless device users, e.g. location tracking and/or prediction) to more cost-efficient and reliable ways of monitoring industry and device/application (type) usage trends. The conceived server arrangement is preferably designed so as to be easily functionally connectable to data input elements such as network-operable wireless devices and other entities via

compatible data interfaces, which makes it particularly applicable in various kinds of situations wherein behavioural, technical and/or contextual information that is based on corresponding raw data as supplied by wireless devices may be utilized.

The server arrangement is enabled to autonomously process and analyze the raw data obtained from wireless devices, while understanding the nature and typical flow of data, and is optimized for handling of such transactional data with various special characteristics including the contextual nature thereof, the data initially being private and user-specific, thus facilitating the calculus of e.g. user-specific behavioural and/or contextual vectors with increased accuracy. As a result, the server arrangement is capable of managing the cumulative intelligence arising from the raw data and utilized data mining algorithms, serving as a clearing house for the data, making sense out of the data, and storing and processing the data in an optimal and technically scalable way, thereby providing a completely different approach for addressing the associated problem than the earlier solutions. Advantageously the server arrangement produces a variety of derived information, such as derived data tables and/or databases, so that upon receipt of a data query the related answer may be briefly provided either by providing the derived information as such or by rapidly turning it into a desired form first.

Secondly, the embodiments may be designed so as to enable handling both intermittent and continuous data transmissions arriving from a plurality (e.g. tens, hundreds, thousands, or even more) of wireless devices substantially simultaneously. More specifically, data is first processed in a raw-level format after which it may be stored e.g. in a temporally organized database, on the basis of which data mining procedures, including e.g. aggregation, such as averaging or producing different frequency or count figures, are configured to calculate various kind of derivative data tables and/or summary statistics, for example, which are typically more useful to the users of the arrangement than raw data. The applied data analysis technology is preferably automatic in contrast to manual labor, being able to run both in a continuous manner (e.g. data input and pre-processing) and discretely (more advanced analysis), optimally leveraging the whole range of data arriving in the arrangement and keeping statistics up-to-date. The data mining engine and the overall arrangement are scalable by design and instead of consisting only of one module, there may be specific modules for different types of tasks. Due to the several layers of intelligence (data input, storage, and mining), modular structure, and the built-in data distribution API, the overall arrangement is easier to scale up than prior art solutions.

Embodiments of the invention suggest using behavioural metrics that are specific to wireless devices, which can be also completed with technical and contextual data. Behavioural metrics convey information, instead of mere data, thus providing more value to the applications of the whole platform. Especially, the behavioural metrics may include usage intensities for the applications and/or other features of the wireless devices, usage frequencies for the applications and/or other features of wireless devices, and/or locus of usage metrics (for example, contextual statistics) for the applications and/or features of wireless devices, and/or stickiness and adoption rates of the applications and/or features of wireless devices. The metrics may also be optimized when integrating the obtained information repository with practical applications, such as mobile advertising platforms, or network management platforms, such as network resource allocation systems. The enrichment, aggregation, e.g. averaging, procedures may be tailored in accordance with the nature of the transaction data available from wireless devices, and the data mining engine advantageously utilizes a number of add-on elements such as application directories (databases) and contextual databases, which may be required in the reliable calculation of the metrics. The data mining engine is preferably self-learning on the basis of behavioural heuristics. The engine also works across different wireless device platforms due to the preferably continuous learning/adaptation and/or standardized data input, instead of being just one more extremely generic or case-specific data mining tool.

Still further, different embodiments of the utilized database structure, i.e. data storage, may be optimally designed so as to enable data processing on multiple layers. For example, raw-level data may be aggregated independently of the higher level averaging and/or other aggregation procedures that utilize already calculated derivative elements such as behavioural metrics, for example.

Yet, various embodiments of the data mining engine may include, in addition to other applicable processing algorithms, a number of different statistical algorithms which are used to identify patterns and/or extract other potentially meaningful information out of data, for instance. These analysis algorithms may include, for example, a number of clustering algorithms for behavioural segmentation of subscribers, factor analysis approaches to categorize applications and content, and/or correlation analysis procedures to provide relevant information to pattern recognition engines.

Last, various embodiments of the data distribution layer of the present invention may provide a unique approach to modularize the use of the data extracted from wireless devices, and to integrate the data mining system with different practical applications. The defined API facilitates independent processing and analysis of data within the arrangement, and optimized provisioning of the data, either in a push or pull mode, to external systems and entities (which may still be physically integrated with the server). These external systems and entities might be, for example, mobile advertising platforms or network management platforms.

According to another aspect, there is provided a method to cultivate observation data for wireless devices, comprising: obtaining and storing the observation data from the wireless devices, the observation data including application usage data; to identify temporally adjacent applications to generate usage metric data; and transmitting advertisements to the wireless devices based on the usage metric data.

In an embodiment, a method for cultivating observation data provided by a number of wireless devices, comprises

-obtaining and storing observation data from a number of wireless devices and optionally supplementary data from a number of other external data sources, the obtained data including behavioural, contextual, and optionally technical raw data relative to the wireless devices,

-establishing and storing a number of derived data elements on the basis of processing and analysing the obtained observation and optional supplementary data, said processing and analysing incorporating aggregating and optionally correlating, clustering, and/or factoring procedures, wherein at least one derived data element includes usage metrics with contextual and optionally technical dimension relative to one or more applications or other features of one or more wireless devices and users thereof, and

-providing derived data from one or more derived data elements to one or more other entities, such as serving a data query created by an external entity through provision of derived information from one or more derived data elements according to the query parameters.

In view of the terminology generally applied in this document, "behavioural data" is related to actual usage habits and behavior of the end-users of the wireless devices. Behavioural data may reflect the usage of applications, communication patterns, content consumption and/or interaction with the device user interface, for example. In this specific context the behavioural data supplied by the wireless devices is also rather specific to particular platforms (e.g. available smartphone device platforms, such as Google Android, RIM Blackberry, Windows Mobile, Symbian S60, Apple iPhone) and in that way incomparable to behavioural data acquired from other

sources. Technical data from wireless devices may include, for example, measurements of cellular tower signal strengths, throughput rates in data sessions, and voice call signal-to-noise ratios. Technical data obtained from wireless devices may naturally complement or overlap with behavioural and/or contextual data. For instance, used wireless device platform may fit under both technical and contextual data. Contextual data provided by wireless devices may further include tags of location, time (i.e. temporal context), surrounding device IDs (e.g. Bluetooth and WiFi devices), and/or device status (e.g. backlight and battery status), for instance.

A "life feed" may refer to any information that may be generated in response to monitoring people's everyday life, including, for example, data on locations, movements, activities and calendar entries. In addition to various actions and events, a life feed can also incorporate user-generated content, such as blog entries and photos. A mobile life feed may be defined as a life feed that can be generated from data provided by a wireless device. As deliberated hereinbefore, contemporary wireless devices and their future embodiments can be considered as best all-around observers of life and thereby also the best automatic generators of life feed. The server arrangement of the present invention is capable of receiving data forming at least part of a life feed and analysing and/or processing it. The analysis results and/or processed data may be provided to third parties, such as mobile advertisers or network management entities, or back to the wireless device or other destination (e.g. social media service) determined by the user of the wireless device, for example.

An API is defined as application programming interface, being e.g. an interface provided by one software module to other modules, typically built for the function of distributing data. An API may support, for example, queries by other system in response to which it supplies data in accordance with the query details. APIs may also be used to define the communications and interoperability between modules of a single system.

A "(mobile) agent" is defined as a set of rules, e.g. at least one application, in a wireless device, capable of preferably seamless and automatic, i.e. non-intervening and non-disrupting other applications, execution on the background. The agent is enabled to perform operations, and communicate with external entities, such as the Internet, or with other applications.

"Observers" are defined in this context as processes capable of generating data items, based on e.g. queries and use of the wireless device's operating system capabilities. Observers are functionally and sometimes also physically sensors, which may automatically sense, for example, changes identified in a cellular base station usage (when the device jumps from the coverage of one tower to the next, for instance). Observers may also refer to channels of user-generated content (for example, blog entries).

"Triggers" are rules and processes that trigger (induce) a certain action. In particular, they may define how the observations can be more effectively and automatically be done in wireless devices. Triggers can be based on time intervals, contextual changes and observations, external requests, or internal requests e.g. in a situation in which more data is needed for some other data points.

The concept of "intelligence" is used in this document in referring to a set of rules, algorithms, databases and/or processes that coordinate the overall procedure or individual micro-processes (for example, the triggering logic) of the associated entity. Intelligence is something that makes the related system to work smarter, in a more optimal way, saving energy and improving accuracy, for example. It may be based on fixed and/or self-learning, adaptive algorithms as well as on external input.

A "server" generally refers herein to a node or at least a logical aggregate of several nodes present in and accessible via one or more networks, for example the Internet. The server may serve clients, e.g. mobile agents running in wireless devices and other entities such as various network services. Clients may thus communicate with one or more centralized servers. Client-server architecture is a commonly used topology of building systems in the Internet.

The concept of "processing" is used in this document to refer to various kinds of actions that may be performed for data. These include data conversions, transformations, formulations, combinations, mash-ups enrichment, correlations, clustering, factoring, normalizing, and/or filtering, among others. Some forms of processing may be actively used in various embodiments of the present invention, including combinations and mash-ups (linking data points together and building relational data structures, for instance), conversions (generating, for example, meaningful streams of information entities from raw-level, unsorted data items, such as observed location points), enrichment (for instance, adding metadata and making

the data richer than originally) and/or filtering (leaving out data that is not relevant or needed anymore, for example).

A "smartphone" is defined in this document as a wireless device capable of running an operating system facilitating installation of add-on applications and enabling a packet data connection to a target network such as the Internet.

"Viral advertising" or "geo-social recommendations" are in this document used with reference to advertisements or other pieces of corresponding information that users can send to other users, or which may be directly sent to (the wireless devices of) target people by the server arrangement of the present invention (or by a further entity obtaining data from the server arrangement) as tied to a certain context, location, and/or suggested automatically to users to be sent to other users by them, or bundled to other contextual pop-ups in an attractive way.

"Audience measurements" and "behavioural research" are used herein to describe the approaches in which usage habits and consumption behaviour of people are measured for the purpose of understanding patterns of usage for a particular platform, such as TV, radio, Internet, or wireless device platforms.

The expression "a plurality of" refers herein to any integer starting from two (2), e.g. two, three, or four.

The expression "a number of" refers herein to any integer starting from one (1), e.g. one, two, or three.

The expression "data transfer" may refer to transmitting data, receiving data, or both, depending on the role(s) of a particular entity under analysis relative a data transfer action, i.e. a role of a sender, role of a recipient, or both.

In one embodiment and practical use case of the present invention the server arrangement is configured to provide valuable data to network and/or wireless-device (local) services and/or applications that want to adapt to user behaviour and context. For example, the arrangement may be utilized for automatically informing the (local) wireless device of a user of the particular behavioural segment of the user, making it subsequently possible for the device to provide personalized use experience, for example, via highlighting useful applications in the menus of the device. For active multimedia users certain new multimedia features can be

automatically suggested or highlighted in the user interface, for instance. The arrangement may provide adaptation data to external entities on the basis of internal triggers, e.g. a timer and/or a triggering logic, and/or in response to queries.

Another, either supplementary or alternative, embodiment and practical use case of the invention relates to social media or other services deployed in computer networks, in which people nowadays spend a lot of time. By being able to analyze the contextual patterns with the help of the server arrangement of the present invention, e.g. the current location, and/or behavioural patterns, such as music consumption activity, the user interface of the service can be automatically customized to reflect such information available of the user, thus facilitating automated mass customization.

In a further, either supplementary or alternative, embodiment and practical use case of the present invention, the present server arrangement may be utilized as an add-on feature in a social media service that provides end-users with a possibility to share content and interact with other people. The behavioural statistics, such as the average distance one travels every day, or number of (music) artists one listens to during a typical month, are useful statistics provided by the arrangement to the user of a wireless device in order to enrich and complement the typical content people share in those services.

Still in a further, either supplementary or alternative, embodiment and practical use case of the present invention, mobile advertising is the selected target application. Preferably the advertising shall be highly targeted, timely, contextual, and/or behaviourally optimized to suit to a particular recipient of advertising. With the present invention, mobile advertising may be implemented with any of such requirements. In addition, due the nature of the invention, as being centralized and supporting data provisioning through a preferably scalable API, the suggested arrangement may be at least functionally integrated with external advertising systems.

In addition, explicit examples of methods to execute the abovementioned tasks in an automated fashion in the server arrangement are disclosed herein.

BRIEF DESCRIPTION OF THE DRAWINGS

In the following, the invention is described in more detail by reference to the attached drawings, wherein

Fig. 1 illustrates the general concept of the present invention from a functional standpoint.

Fig. 2 illustrates different features of an embodiment of the present invention via a more technical approach.

Fig. 3 is a combined block and flow diagram of one embodiment in accordance with the server arrangement of the present invention.

Fig. 4 is a combined block and flow diagram of one embodiment of a data input interface applicable in the server arrangement.

Fig. 5 is a combined block and flow diagram of one embodiment of a data input interface applicable in the server arrangement.

Fig. 6 is a combined block and flow diagram of one embodiment of data storage applicable in the server arrangement.

Fig. 7 is a combined block and flow diagram of one embodiment of a data distribution API applicable in the server arrangement.

Fig. 8 represents a flow diagram of a method in accordance with the present invention to be performed by the server arrangement.

DETAILED DESCRIPTION OF THE EMBODIMENTS

Reverting to the foregoing and with particular reference to figure 1, the overall architecture in connection with the present invention is illustrated first in a conceptual sense. The disclosed system is divided into two parts, namely a mobile (agent) part executed in one or more wireless devices 102, 104, 106 of users 102b, 104b, 106b, respectively, and a server arrangement part 112 comprising one or more server devices functionally arranged so as to establish a server entity of the present invention. The users 102b, 104b, and 106b are each living their life and using their (personal) wireless devices 102, 104, 106 in different environments and scenarios, which may be monitored via the wireless devices 102, 104, 106 by the applied, either commonly available or proprietary, solution for mobile data acquisition. The server arrangement 112 is typically connected to a communications network 110 whereto also the wireless devices 102, 104, 106 are provided with access e.g. via one or more access networks 108a, 108b, which may be cellular or wireless local area networks, for instance. External entities 114 such as services/servers may be

connected to the server arrangement 112 via the network 110, for instance, for obtaining data from the server arrangement 112 and/or for providing supplementary data, note arrow 118, such as weather, location or other contextual information thereto. Rectangle 116 illustrates an example of data provided by the server arrangement 112 to one or more external entities 114 for use in different applications. The data is derived from the raw data, as supplied by the mobile agents of the wireless devices 102, 104, 106 and optional other entities, in the server arrangement 112 according to the principles described herein. The raw data may be subjected to different processing, factoring, aggregating, clustering, enrichment, filtering, etc. algorithms as a result of which the derived data is obtained. The derived data may represent figures relative to, e.g. one user or a group of users selected as desired, wherein the figures may include intensity frequencies or usage frequencies of associated actions or events (as shown at 116), for instance. The derived data may be visualized numerically/textually, e.g. via charts, and or graphically, e.g. via graphs.

Different embodiments of the present invention are thus partially enabled by the wireless devices 102, 104, and 106, which preferably automatically collect, via software and optionally hardware (e.g. sensors) sensing functionalities, a considerable amount of behavioural, technical and/or contextual data, i.e. observation data, and further automatically transmit at least part of the gathered data to one or more servers 112 at optimal time instants. Mobile agent software may be used for this purpose. The wireless devices 102, 104, 106, such as smartphones or communications-enabled PDAs (personal digital assistant), capable of providing data to the server arrangement 112 may especially comprise an observation logic for conducting observations relative to the wireless device and changes in the observation environment thereof, such as an occurrence of a certain event or action, and a data handling logic for storing, optionally pre-processing, analysing and/or aggregating observation data points of the gathered observation data to timed observation data transmissions via wireless transceiver towards the server 112. The observation logic may be arranged in each wireless device 102, 104, 106 to run on the background as intelligently triggered and mostly passively with minimum battery consumption and capacity usage, but still with comprehensive data acquisition capability relative to the relevant data points regarding the data associated with the device. The events may include, for example, substantially non-user-initated incidents, such as battery status or location (e.g. cell tower or GPS-identified) change, not at least directly initiated by the user of the device. The actions may include substantially user-initated intentional activities and incidents,

for example use of the web browser, movements, reading a message, etc. Some incidents may be also considered to conveniently fit both of the above incident classes.

As a tangible, though purely exemplary, use case, upon recognizing that the active base station has changed an associated observation can be made so that the details of the base station with the desired parameters like signal strength and/or timing advance will be scanned. Accordingly, a related data point may be written into a data log. The observation logic may collect data points based on communication actions (for instance, initiating a call, answering a call, sending a message such as an SMS (Short Message Service), MMS (Multimedia Message Service), or e-mail message, receiving a corresponding message, etc.), sensor data (e.g. temperature, acceleration, position (orientation and/or location via e.g. GPS (Global Positioning System)/cell identification/triangulation), light intensity), application usage, microphone usage, loudspeaker/audio output such as music reproduction, camera usage, any user input or action in general, calendar entries (additions/deletions and/or actual realization/activation thereof), and in principle the observations can range from simple temperature-type logging to all-day audio and video recordings, for instance, which are automatically observed (~recorded) in the device.

Preferably the wireless devices are configured to push the data towards the server at appropriate time instants. For example, related intelligence (logic) embedded in the device, may determine the optimal time to transmit the processed data from devices to server(s). Such intelligence may be implemented using contextual triggers (e.g. location changes), behavioural triggers (e.g. user's actions), time limits (e.g. regular transmission), emergency transmits (for instance, when fire observed in the proximity of the phone), cost-efficiency (transmitting after a certain threshold of data is collected to local memory, limiting the costs of transmission) and battery-optimization (saving as much of the battery as possible). The intelligence logic is preferably capable of learning from the collected data, observed patterns, and/or input from external entities such as servers, and adapt to contextual dimensions as well.

The wireless device according to any embodiment of the present invention comprises at least one wireless communications transceiver. Non-limiting examples of the transceivers include a GSM (Global System for Mobile Communications) transceiver, a GPRS (General Packet Radio Service) transceiver, an EDGE (Enhanced Data rates for Global Evolution) transceiver, a UMTS (Universal Mobile

Telecommunications System) transceiver, a WCDMA (wideband code division multiple access) transceiver, a PDC (Personal Digital Cellular) transceiver, a PHS (Personal Handy-phone System) transceiver, and a WLAN (Wireless LAN, wireless local area network) transceiver. The transceiver may be such that it is configured to co-operate with a predetermined communications network (infrastructure), such as the transceivers listed above. The network may further connect to other networks and provide versatile switching means for establishing circuit switched and/or packet switched connections between the two end points. In addition/alternatively the device may comprise a wireless transceiver such as a Bluetooth adapter meant for peer-to-peer communication and e.g. piconet/scatternet use. In addition, the terminal may comprise interface(s) for wired connections and associated communication relative to external entities, such as an USB (Universal Serial Bus) interface or a Firewire interface.

Figure 2 illustrates various technical aspects of the present invention and related wireless devices in light of a certain feasible embodiment. A data source of the server arrangement 112 such as a wireless device 102, 104, 106 is typically provided with one or more processing devices capable of processing instructions and other data, such as one or more microprocessors, micro-controllers, DSP's (digital signal processor), programmable logic chips, etc. The processing entity 220 may thus, as a functional entity, physically comprise a plurality of mutually co-operating processors and/or a number of sub-processors connected to a central processing unit, for instance. The processing entity 220 is configured to execute the code stored in a memory 222, which may refer to instructions and data relative to the mobile agent software architecture 206 for providing the server arrangement 112 with observation data. Software 206 may utilize a dedicated or a shared processor for executing the tasks thereof. Similarly, the memory entity 222 may be divided between one or more physical memory chips or other memory elements. The memory 222 may further refer to and include other storage media such as a preferably detachable memory card, a floppy disc, a CD-ROM, or a fixed storage medium such as a hard drive. The memory 222 may be non-volatile, e.g. ROM (Read Only Memory), and/or volatile, e.g. RAM (Random Access Memory), by nature.

The UI (user interface) 226 may comprise a display, and/or a connector to an external display or data projector, and keyboard/keypad or other applicable control input means (e.g. touch screen or voice control input, or separate keys/buttons/knobs/switches) configured to provide the user 102b, 104b, 106b of

the device 102, 104, 106 with practicable data visualization and device control means. The UI 226 may include one or more loudspeakers and associated circuitry such as D/A (digital-to-analogue) converter(s) for sound output, and a microphone with A/D converter for sound input. In addition, the device 102, 104, 106 comprises a communications interface such as a radio part 224 including a wireless transceiver for general communications with other devices and/or a network infrastructure and optional other wireless or wired data connectivity means such as one or more radio transceivers or wired interfaces (e.g. Firewire or USB (Universal Serial Bus)) for communication with other devices such as terminal devices, peripheral devices or network infrastructure(s). It is clear to a skilled person that the device 102, 104, 106 may comprise numerous additional functional and/or structural elements for providing advantageous communication, processing or other features, whereupon this disclosure is not to be construed as limiting the presence of the additional elements in any manner.

Correspondingly, the server arrangement 112 may comprise one or more computer devices 234 comprising a communications interface 254 such as a LAN (Local Area Network) adapter, e.g. Ethernet adapter, a processing entity such as at least one processor 250 for processing data, a memory 252 for storing data, server side software architecture 258 and UI 256. Different embodiments of the server arrangement internals are described in more detail hereinafter.

Software functionalities 206 and/or 258 may be implemented as one or several, mutually communicating, software applications executed by the processors 220 and 250, respectively. Either computer software (product) may be thus provided on a carrier medium such as a memory card, a memory stick, an optical disc (e.g. CD-ROM or DVD), or some other memory carrier. The instructions required for implementing the application(s) may be stored in the carrier medium as executable or in some other, e.g. compressed, format, such that the software may be transported via the carrier medium to a target device and installed therein, e.g. in the hard disk thereof, or executed directly from the carrier medium in the target device by loading the related instructions to the memory 222, 252 of the target device not until execution, for instance. Alternatively, software 206, 258 may be transmitted to a target device over the air via the wireless transceiver or a through a wired communications connection.

Figure 3 represents a combined block and flow diagram of one embodiment of the server arrangement in accordance with the present invention. As explained

hereinbefore, a mobile agent may observe events, actions and/or properties in each wireless device and optionally perform pre-processing for the observed data and generally manage the wireless device-based intelligence regarding data handling and transmission to the server arrangement operated by one or more at least functionally connected server apparatuses. On the server side, a data input interface 300 is responsible for receiving and preferably pre-processing behavioural, contextual and/or technical data collected by the wireless devices, and for storing data in one or more raw level databases by providing the data to a data storage module 400. The data input interface 300 may also receive instructions and/or updates regarding e.g. how to handle data, from a data mining engine 500, which is illustrated by an arrow between the engine 500 and the data input interface 300. Data storage 400 may handle both raw and derived data, i.e. it preferably stores all the raw data as processed by the data input interface 300 and additionally further manages the data processed and analyzed by the data mining engine 500, i.e. the derived data. The data storage 400 maintains a so-called data warehouse, which may basically store the relevant information including derived data to be later provided to external entities through a data provisioning and distribution API 600. The data storage 400 also contains other required data tables, for example mapping tables for applications, country codes (MCCs), and operator codes (MNCs), which may be updated dynamically. The data mining engine 500 may process the raw data provided by the data storage 400, and supply the processed data and analysis results, i.e. derived data, back to the storage 400 after each analysis round, for example.

In figure 4, a combined block and flow diagram of one embodiment of a data input interface (module) 300 is depicted. The parser 301 receives data such as data logs from external systems, typically from the mobile agents installed in the wireless devices, or via intermediate or other systems such as data servers, and preferably reads different kinds of data log formats (for example, some of the data can be defined through the XML specification), while optionally also decompressing and/or decrypting data, when needed. Advantageously the parser 301 may further detect and leave out, i.e. filter out, corrupted data by monitoring e.g. data values (do they follow a predetermined range, for example). The data input interface 300 is guided by an intelligence engine 305 that maintains the knowledge regarding rules and processes by which the data input interface 300 operates, i.e. associated logic. The intelligence engine 305 may be controlled by a data mining engine as explained hereinafter and illustrated by the arrow "Rules". The parser 301 may assign unique user identification numbers for each set of incoming data, i.e. basic identifiers for the incoming data. This facilitates more efficient processing of data later in the

system, as only such user identification codes need to be used in referring to a particular user's data points. A pre-processing component 302 may perform, preferably in a continuous manner, operations on the data that is fed to it by the parser 301. Pre-processing may, for example, include enrichment, such as addition of cell tower indices to incoming data points, and/or sorting of data points (if the data is not already received in a chronological order). A dynamic statistics module 303 may, substantially upon data arrival, derive and/or update simple and straightforward statistics out of the raw data flow, thus updating, for example, the status of each user stored in the system (for example, updating at the time of receiving a data point that such data is the most current for the respective user), thereby keeping track for example for how many people some data is received during the past, e.g. 12, hours. Dynamic statistics may typically be simple counters for administrational purposes, and no advanced aggregation or statistical calculus is needed (in contrast, derived data points may utilize historical data and aggregation procedures). Dynamic statistics may also be used as triggers for certain activities (like automated data exports by the data distribution module). A raw data output module 304 may handle the communication of raw data to a data repository of a data storage module to be reviewed next.

In figure 5, a combined block and flow diagram of one embodiment of data storage 400 is depicted. Input of dynamic statistics entity 401 receives the brief updates sent by the data input interface 300, such as status of individual users (or devices) providing data to the system. These data points may be practically immediately updated in a data warehouse entity 403, which at least logically maintains the higher level data instead of raw data. A different data stream, though preferably (pre-) processed by the data input interface 300, is received in a raw data repository 402. The raw data repository 402, may preferably store all raw data and index data efficiently by using e.g. masking identification numbers calculated over user identification numbers, being thus able to use physically separate databases in storing the vast amount of private data.

By using masking and utilizing optimal database architecture, private data if collected (such as email addresses and names) and research data (behavioural, technical, contextual data) can be separated. Only research data may be needed in analytics. Indexing, buffering, replication and/or other database configurations can be adjusted also for the requirements of the data mining engine and back-up purposes. Effectively the database design facilitates optimal use of memory capacity together with optimizing latency and other important factors in using the data. The

raw data will be provided to a data mining engine upon request and received back as processed, i.e. derived, data and analysis results for storage in a data warehouse 403. The data warehouse 403 may therefore maintain cultivated information, i.e. data that has been processed to a form that can be utilized in practical applications right away, or to a form that is quicker to process further with desired algorithms, such as clustering, or statistical procedures, such as averaging or other type of aggregation. The data stored in the data warehouse 403 may be optionally provided to a data distribution and provisioning module through a separate module, data output 404, which may be configured to take care of the caching, buffering and/or scalability of data export actions. In many cases, the data warehouse 403 may be physically distributed across several server apparatuses.

In figure 6, a combined block and flow diagram of one embodiment of a data mining engine 500 is depicted. Raw data may be first stored in a raw data module 501, e.g. in a set of databases, just to temporarily store the data arriving from the data warehouse for analysis. Accordingly, module 501 works as a buffer, loading data optimally to data processing functions. A data processing module 502 fulfils a number of specific processing and analysis needs in view of behavioural, contextual and/or technical data obtained from the wireless device platforms. For example, the processing of application session logs may be one rather concrete and in many cases complex process taken care of by the data processing module 502. Associated processing, which relates e.g. to the time-orientation of data (e.g. application of a certain process rule on a dataset of chronological order), or otherwise replication or adjustment of certain required pre-processing activities that are not yet done on the data, may be performed by the data processing module 502. A data analytics module 503 may, in contrast, perform operations such as averaging, other aggregation, correlation, clustering, factor analysis etc., which are not micro-level processes. The data mining intelligence engine (logic block) 504 may guide both data processing and data analytics, and additionally also the data input interface. As the intelligence engine 504 is preferably centrally located, it is easy to update so that changes may be adopted substantially immediately in the analysis procedures. The processed data is transmitted back to the data storage, and further to the data warehouse thereof, via a processed data module 506, for example. The data mining procedures can be run, for example, for all the data received yesterday, but they can also be run for larger patches of data (for example to process all the raw data in the raw data storage) if indicated in the parameters when it is launched. The data mining procedures may be automatically revoked to process and analyze for

example the past three days' of data every night, to automatically keep analytics data updated.

In Fig 7, a combined block and flow diagram of one embodiment of a data provisioning and distribution API 600 is presented. A data distribution control 603 may maintain the configuration of the module, meaning, for example, the rules on how to transmit and provide data, and optionally e.g. privacy settings, optionally including certificates, regarding the data handling relative to individual external systems. Triggers 602 may be configured to guide a data query module 601. The triggering logic follows the logic assigned by the control module 603, for instance, whereupon triggering of data queries is enacted either actively (data is polled from the database(s) actively), or by redirecting and coordinating external requests. The data query module 601 manages the interface with the data repository, directing the data to a data provisioning module 604, which maintains a variety of interfaces needed in view of data provision to external entities via their data interfaces, such as mobile advertising platforms.

Next, some embodiments of the applicable processing for the raw level data, which is stored in one or more raw-level databases, are explained. The data input interface is advantageously configured to operate smoothly and efficiently, being, for example, able to input data in a standardized format, such as XML (eXtensible Markup Language) format, from several wireless devices substantially simultaneously thanks to the distributed, independent structure of data input processes and associated buffering. The data input interface may read received data log rows of wireless devices one by one, for instance. To support the scalability and/or interoperability of the overall arrangement, the data input interface may be configured to recognize familiar application names and application identification numbers, for example, right in the parser. The parser may be responsible for real-time data input processing from external systems, understanding the form, structure and/or content of input data. If the application is known (the name or application ID number recognized and categorized), renaming (harmonization of application naming) and/or categorization may be performed already in the parser, before anything is stored into the database. Effectively, in this example, applications can be identified by name and/or by distinct application identification codes. Separate application mapping files may reside in the server memory, and they can be matched with incoming data and the presented identifiers (names and application IDs) to execute processing (renaming) and to add metadata (application categories).

The mapping files on the server can be preferably updated dynamically, either automatically (with interface(s) to external systems), or by hand.

The parser (and/or the pre-processing block) may also keep contextual information in the buffer when reading data, e.g. the user's location, and attach contextual metadata to observations that are ready consequently, being able, for example, to map the name of a previously launched application into the data, thereby identifying for example the differences between the execution (starting) times of applications (which can be used in calculating the duration of application sessions, for instance). The parser, which may advantageously analyze several data points, e.g. data rows, substantially at the same time in its memory, also enables to identify duplicate data or erroneous data feed items (that can be identified by monitoring several consequent rows). It may further perform contextual enrichment in real-time. The parser may further be made responsible for raw-level filtering and processing of data preferably done at the time of storing data into the raw data repository. These activities may include, for example, removal of certain types of data points. The rules of processing can be stored in the intelligence engine of the data input interface and be changed preferably dynamically.

Some tangible embodiments of data mining activities of the present invention are disclosed. In particular, it is explained how the application session logs can be processed so as to make data more reliable and meaningful. Application session logs are rather important piece of behavioural source data. One challenge is that there are many kinds of data points available. The data collection systems, e.g. the ones in the wireless devices, may log application process starts and stops, and separately application activations and minimizations on the user interface. In addition, all different wireless device platforms may have unique names and application identification codes for the same or similar applications. In addition, naming of applications also depends on, for example, the language of the device. By taking the example of application logs, the data processing engine of the invention is able to recognize, based on e.g. fingerprinting (sensing the incoming data and for example associated IP packet header information), the particular wireless device platform (for example Windows Mobile, Symbian S60, or Apple iPhone) that is supplying the data, to build meaningful data items out of raw data (for example with application start and stop items to build a new application session item including both the start time and duration), and to map all application occurrences with coherent, universal identification codes, handled in a centralized way by the platform, even in post-analysis. If new, still unidentified, names of application are

observed, there can be a separate request initiated for developers (through email, for example) to map those applications into the existing database .The other choice in application processing is to perform all or some of these activities in the parser. The advantage of carrying out processing in the data processing module of the data mining engine (not when receiving the data) is the fact that if the information in the mapping tables, for example, is changed, or if more complex procedures (such as combining already calculated behavioural indices with transactional data to perform normalization) need to be done, then the data processing engine can be used to run even patches of already stored data and spend more time with the procedures. By definition, the parser may work only at the time of receiving data and is not initiated to process data that is already in the database. Because of the real-time requirements of the parser implementation, it may be much simpler as to its implementation than the data processing engine.

The data processing engine may be configured to compare, for example, execution initiation times of applications and/or to calculate the run times of applications. For some applications, like music players, process start and stop times should be used for the calculation of session durations, because music player is typically left on the background without any separate user interface level activity. For most other applications, for example calendar and document viewing applications, the nature of the application states that it is natural to measure duration from user interface level activation logs.

The data mining engine and especially e.g. data processing engine thereof may, for example, exclude screensaver, key lock and menu processes from the dataset, and then build more precise session-level variables, for example face time or reliable run times of applications, without letting system processes and other disturbing data points to bias results. In practice the systems utilizes black lists in certain procedures, which helps in streamlining and filtering, for example, application or Internet usage (URL) logs. The data processing engine may also able to monitor, for example, consequent SMS or other messages, to group received messages from a particular contact with an outgoing SMS or other message to the same contact, thus identifying chat-oriented sessions and/or reply messages. In addition, for example opening, reading and/or removal times of SMS or other messages can be mapped through heuristics to the time of receiving the message, facilitating multiple kinds of duration-oriented statistics. Yet, for example, receiving multiple e-mail or other messages during a certain time period can be mapped to a specific session of the

messaging application, thereby making comparisons both on an application session level and micro-level messaging possible.

Third, some examples of the analytics procedures are elaborated relating to discrete aggregation procedures. In particular, it is explained what kind of insights can be derived by just aggregating, such as averaging, statistics provided by the processes described herein. One major focus of data aggregation may be in calculating intensity variables, such as activity measures measured by the corresponding usage time, e.g. minutes of usage, per selected time unit (e.g. hour, week, month, etc.), number of sessions per time unit, and/or number of events or actions per time unit. These variables may be calculated for each user, a plurality of users, for each application (for example, a calendar application), for each application category (for example, email applications), and/or a device feature (for example, USB connections) category, across contexts, through powerful aggregation procedures. In order to facilitate scalability, the recurring processes of the data analytics procedures may first calculate intensity variables on the level of individual actions, which can be used as source data for higher level averaging. The intensity variables may be stored in the data warehouse, e.g. one data row describing a user identification number, date, time, and/or details of the action/event (for example the name of an application, such as calendar or voice call, its identification number, categorization information like application classification group, and session-level details, like duration of a session or length of a text message). These more micro-level derived data entities may be processed frequently (for example, every night) and/or on a need basis. With a scalable aggregation engine, which may be run on the basis of parameters such as types of data points needed, user identification information, and range of dates, either regularly or through separate triggering logic, the existing data can be processed and transformed into new data entities such as tables, for example one presenting the number of calendar application activations for each user for each day. By aggregating the data even further, for example, this data table can be processed to identify the number of distinct users using calendar at a certain instant such as on a specific date, or alternatively, for each user an arithmetic mean number of calendar application launches can be calculated for the period under study. Because the micro-level data is in such a harmonized, though precise, form, multiple kinds of aggregation, such as averaging, methods can be used in a scalable fashion to derive meaningful statistics of usage behavior and contextual patterns. Some practical behavioural application-level individual statistics besides usage intensity figures, that can be derived from raw data, include for example usage frequencies of applications (during an average month, on how

many distinct days the application is used, for example) and significance of application (out of a particular user's total time spent with the device, how much time goes on a particular application on average, for instance).

Fourth, some practical explanations regarding the possible structure of used databases is provided, explaining how the scalability can be supported therein. The data stored in the databases, in the raw-level format, is preferably, however not necessarily, in Unicode format, and various relations may be utilized in storing the data optimally. For example, a typical way to handle wireless device application information is to attach a unique application identification number for each of the application process execution start data rows, which are contained in the raw-level data storage. Separate databases may be established to contain metadata related to application identification codes, including for example name, application type, and application category information. Another example comes from the storage of contextual information. For example, for each date and hour, and for each user, a variable can be calculated identifying the country of presence, and possible some metadata such as whether that country is the user's home country or not. By simply matching raw-level data points with this separate table of context data, contextual information can be attached to raw level data points. Such derived data table may be used, for example, to aggregate the total application face time across application categories separately for different contexts such as home and roaming contexts. In the database architecture, the raw level sessions and activity logs, and most important mapping files, are preferably kept separate, or at least separable, from the derived data tables. Derived data (tables) may be calculated from raw-level data (tables), utilizing aggregation incorporating functions like averaging. For example, the derived data tables, in the context of mobile application usage, may include a data table containing average daily total face time, given e.g. in minutes, with the device in view of each user and each date, or a data table communicating the average number of distinct applications used by each user during a time period such as each calendar week.

Derived database entities such as data tables are typically calculated based on the raw data, being dynamic in the sense that more recent data may, in many occasions, be added to accompany (and added to the "end" of) older data, although sometimes the older data may be at least partially replaced, for example, if e.g. time dimension is not needed in the table. Derived data tables contain valuable higher-level information, instead of mere raw data, in the sense of conveying something meaningful, like behavioural statistics, required by external systems and practical

applications. In addition, derived data entities such as data tables and some of their content may be output to visualization engines in a preferred, e.g. standardized, format to facilitate visualization and representation of information. Derived data tables represent behavioural key performance indicators (KPIs), for example average time spent with certain applications per unit of time (e.g. usage minutes per week), or average stickiness (e.g. relative proportion of application trial users continuing usage actively) or adoption rates (e.g. relative proportion of people interested in application(s) actually using them).

In the following, exemplary implementation guidelines for factorization and clustering of subscribers and content are provided. Factorization of behavioural, technical, and/or contextual data means reducing the unnecessary degree of accuracy in the data, while also recognizing patterns across dimensions. For example, a variety of descriptive behavioural statistics can be calculated for every user, including, for example, a number of minutes or other time units a user on average spends with music applications every day or during some other time period, or the proportion of all outbound voice calls taking place abroad. By using factor analysis across a certain number of descriptive statistics, higher-level factors correlating with the original variables can be found, at the same time building higher level understanding of interrelationships and correlation between variables. A practical application is to analyze usage frequency variables of application categories (for example voice, instant messaging, email messaging, music, video, map, imaging, office etc.) and build factors on top of them, for example to derive a generic multimedia usage indices out of music, imaging and/or video application usage variables, completely automatically. In addition to applications, the presented factor analysis, being based on derived data tables, may be applied to different means of communication, type of consumer content (music), or e.g. the type of web sites browsed in the Internet, for instance.

From the perspective of segmentation, a plurality of users can be clustered to form behaviourally coherent groups, for instance. Behavioural segmentation, instead of static segmentation models such as the ones based on demographics or attitudinal (intentions and satisfaction of users) factors, is dynamic and potentially more meaningful approach from the perspective of many target applications including mobile advertising, for example. Behavioural (for example multimedia, data and/or communication service usage) segmentation can be also combined with contextual statistics (for example the amount of movements taking place on the weekends). As the initial stage of clustering, the nodes of the graph may represent e.g. users, and

the weights between the nodes may be Pearson correlation coefficients (or equivalent), calculated, for instance, based on behavioural metrics (for example application usage times per week: browser, voice, messaging and map application usage minutes per week) provided by the used platform. In the examples here, the original data to describe edges between nodes may be multi-dimensional, i.e. multiple weights between nodes can be calculated.

In view of automatic clustering, an embodiment of a modularity-based approach to analyze the graphs produced by the platform is described. The modularity may be defined as follows. Denote by e_{ij} half of the fraction of edges in the graph that connect vertices from community i to community j, given that $i \neq j$. Half of the fraction is chosen instead of the full fraction since the normalization demands that $e_{ij}+e_{ji}$ equals the total fraction. Denote also by e_{ii} the fraction of edges inside community i.

Using this notation, the sum

$$b_i = \sum_{i=1}^{N} e_{ii} \tag{1}$$

equals the fraction of edges that fall within the communities of all edges, while

$$a_i = \sum_{j=1}^{N} e_{ij} \tag{2}$$

is the fraction of ends of edges that emanate from vertices in group i. Now, if all edges were connected at random, the fraction of them inside community i would be a_i^2. This lets one define the modularity Q as

$$Q = \sum_{i=1}^{N} (e_{ii} - a_i^2) \tag{3}$$

If the edges are random, the modularity equals zero, whereas values $Q>0$ indicate a clustered structure. Usually values of about $Q>0.3$ or 0.4 may be considered as signs of significant clustering.

The used method of optimizing modularity works as follows. Let initially each vertex form an own community. Consider all possible aggregations of two communities into one, and compute the modularity after these joins. Choose the one with the highest modularity and aggregate the communities together. Repeat this procedure iteratively for the new set of communities until there is no pair of two clusters the joining of which would increase the modularity. The communities at this point are then the best division of the original graph into communities in terms of the algorithm.

Denote by d_{ij} the measurement of node i in measurement group j. By dividing the values with the measurement group averaged ones

$$\alpha_j = \frac{\sum_{i=1}^{N_p} d_{ij}}{N_p} \tag{4}$$

where N_p is the number of nodes, the scaled measurements β_{ij} can be defined as

$$\beta_{ij} = \frac{d_{ji}}{\alpha_j} \tag{5}$$

leading to the vectors

$$\gamma_k = (\beta_{kj})_{j=1}^{(N_a)} \tag{6}$$

describing the patterns of individual node k. Here, N_a is the number of measurement groups. Using these vectors, it is possible to define the similarity coefficients for nodes k and l as explained below.

Using these similarity coefficients, a fully connected weighted graph may be built with the edge between nodes k and l having the weight w_{kl} determined, for example, by using Pearson correlation coefficient. The algorithm takes the weights of the edges into account. This may be done simply by redefining the factors e_{ij} to

$$e_{ij} = \frac{w_{ij}}{2\sum_{x}\sum_{y>x} w_{xy}} \qquad (7)$$

where the summation is over all pairs of vertices. E.g. Newman's algorithm may be applied to produce a division of the nodes into clusters.

Further, an embodiment of a pattern recognition model to be used in connection with the present invention is explained, making a point on how to facilitate the arrangement's self-learning capabilities. In this example, users' application logs are studied, and in particular the jumps from one application to next are analyzed.

Users may switch from an application to another sometimes fully based on feelings or discrete use cases they want to do with the smartphone. However, in many cases usage sessions are tied together, for example after taking a picture with the camera, a wireless device user may want to send it to their friends through multimedia messaging. By feeding historical patterns of application usage, including information on the identity and/or type of applications, session durations, and/or the identity and type of temporally adjacent application(s), statistical models of usage behavior can be built, based on, for example, Markov models and/or neural networks. With the resulting probabilistic models, reasonable estimates can be calculated for each user's status at any particular time, for example predicting for how long he will still be using the same application, or what is the personal probability of initiating a certain session, e.g. a camera session, after the current activity. Similar estimates may be determined for an aggregate of several users. These kinds of pattern recognition and predictive models, based on behavioural and contextual models together with the data processing capabilities introduced into the embodiments of the present invention, have direct applications in sending predictive advertisements or other personalized data to users, or communicating likely movement patterns (number of people, direction, times, locations) of people to e.g. municipal authorities in advance of actual movements even taking place, for facilitating traffic control, for instance.

Next, few API implementation guidelines are elaborated, regarding the practical integrations to external entities such as different external systems including advertising and social media systems. The data distribution API is preferably based on virtualized and scalable clusters of databases and used with a semantic database

model enabling various kinds of queries, from direct to more complex, semantically formulated ones. The data distribution API advantageously facilitates both pull and push model of data distribution. At least either model shall be anyhow supported. In addition, it centrally manages each user's data, based on universal privacy settings, user-specific settings, data sharing conditions, and/or other centralized data management settings the owner or administrator of the platform has defined. The data distribution API makes sure that the data is funneled efficiently to proper interfaces in a correct format. The data can be output in various formats. Widespread standards such as XML and GeoRSS (RSS, Really Simple Syndication) may be used in pre-defined data streams. In addition, a number of customized interfaces can be easily built between the API and e.g. widely used external services in the communication network by applying the available request formats of the API. Some exemplary data request (query) types include:

1. Specific behavioural statistics (for example, average application, e.g. web browser usage, activity during a time period such as weekends)
2. Status information retrieval (location, context, and/or profile status of a user of a wireless device)
3. IDs of wireless device users currently in a certain context (location, application session)
4. IDs of wireless device users having used the web browser at least once during a certain time period
5. List and/or number of countries a specific wireless device user or a group of users have ever or during a predetermined period visited
6. The most typical location for a specific wireless device user at a certain instant, e.g. during Friday afternoons
7. Likelihood estimates (predictions) for a particular wireless device user's location in a desired time period, e.g. in one hour
8. The image content feed from a particular location during a certain time period
9. Most recent behavioural profile of a wireless device user based on application usage patterns, preferably updated frequently such as every night

Some parameters that can be, either alone or as a desired combination, included as part of the queries include:

1. Wireless device user identification
2. Time indication (temporal context)
3. Location (context) (old, current, future)
4. Status information (e.g. "moving", "busy")
5. Context (semantic, for example home, school, office, bus)
6. Behavioural patterns (application usage)

The data distribution API facilitates external widgets or applications to execute queries in the database. For example, a specific application can be built for social media (networking) or other service, provided with its own user interface and functionalities that plots the data provided by the arrangement of the present invention through the data distribution API. The data distribution API does not have to take a standpoint on how the data is to be used. Rather, it specifies a multi-use interface to easily conduct queries in the intelligent and optimized database(s) of derived data.

The data distribution API is typically built on top of derived data elements such as data tables. Potentially the most widely used practical applications for the API layer include the interfaces towards mobile advertising platforms, to which updated contextual and behavioural profiles of users may be sent on regular intervals, for example every x, the x being e.g. sixty, minutes. Other examples include interfaces built towards social media services and other web services, which may utilize real time behavioural and contextual data in improving the service end-user experience, in building new features, in adjusting the user interfaces dynamically for each user, in tailoring content, and in enriching customer data bases with more dynamic, instead of static data. The purpose of the data distribution API is to leverage the processed and derived behavioural and contextual statistics, which may be updated dynamically based on incoming data streams, and to distribute information to those systems and interfaces in a scalable and flexible manner, thereby maximizing the value of the arrangement.

Figure 8 is a flow diagram of an embodiment of a method in accordance with the present invention.

At 814 a server arrangement in accordance with an embodiment of the present invention is obtained and configured, for example via installation and execution of related software, for managing observation data of wireless devices. At 816 observation (raw) data is obtained from a number of wireless devices. Optionally also supplementary data from a number of other external data sources (e.g. metadata providing location information) is received. At 818 a number of derived data elements are established on the basis of processing and analysing the obtained observation and optional supplementary data, said processing and analysing incorporating aggregation and optionally correlating, clustering, and/or factoring procedures, wherein at least one derived data element includes usage metrics with contextual and optionally technical dimension relative to one or more applications or other features of one or more wireless devices and users thereof. Both observation (raw) data and derived data may be stored in one or more locations at least functionally connected to the server arrangement and forming data storage. At 820 a data query formulated by an external entity is served through provision of derived information from one or more derived data elements according to the query parameters. Additionally or alternatively derived data may be provided to other elements by a push model. The broken line depicts the flexible repeatability of different phases. A skilled person realizes that the illustrated flow diagram is indeed merely exemplary and the nature and number of method steps, not forgetting the mutual order thereof, may be dynamically and/or use case –specifically adjusted.

The present invention thus describes an optimal technical architecture and a related method to automate the conversion of behavioural and contextual data into more meaningful information, such as desired statistics in the person-time-context-application space, depending on each embodiment and use scenario of the invention.

The scope of the invention can be found in the following claims. Notwithstanding the various embodiments described hereinbefore in detail, a person skilled in the art will understand that different modifications may be introduced to the explicitly disclosed solutions without diverging from the fulcrum of the present invention as set forth in this text and defined by the independent claims.

Claims

1. A server arrangement to process observation data of wireless devices, comprising:

data input logic to obtain observation data from the wireless devices the observation data comprising application usage data;

data mining logic to process the observation data to identify temporally adjacent applications to generate usage metric data;

and

data distribution logic to transmit advertisements to the wireless devices based on the usage metric data.

2 The server arrangement of claim 1, wherein the data distribution logic is to generate derived data for personalized advertising for users of the wireless devices based on behavioural characteristics of the user determined from the usage metric data.

3 The server arrangement of claim 1, wherein the data mining logic is to determine an intensity variable based on the temporally adjacent applications.

4 The server arrangement of claim 1, wherein the data mining logic is to utilize start and stop times for the temporally adjacent applications to calculate a usage session duration for a first type of application, and to determine a usage session duration for a second type of application.

5 The server arrangement of claim 1, wherein the data mining logic is to determine a statistical model of usage behavior to provide a status estimate for a user at a particular instant based on the usage metric data.

6. The server arrangement as defined in claim 1, wherein the data mining logic is to build a behavior model based on the identified temporally adjacent applications, the behavior model to describe user behaviors associated with the wireless devices.

7. The server arrangement as defined in claim 6, wherein the data mining logic is to apply the behavior model to predict a usage duration of a second application in response to detecting usage of a first application.

8. The server arrangement as defined in claim 1, wherein temporally adjacent applications comprise at least two applications in sequential time order.

9. A method to cultivate observation data for wireless devices, comprising:

obtaining and storing the observation data from the wireless devices, the observation data including application usage data;

to identify temporally adjacent applications to generate usage metric data; and

transmitting advertisements to the wireless devices based on the usage metric data.

10. The method as defined in claim 6, further comprising building a behavior model based on the identified temporally adjacent applications, the behavior model to describe user behaviors associated with the wireless devices.

11. The method as defined in claim 10, further comprising applying the behavior model to predict a usage duration of a second application based on usage of a first application.

12. The method as defined in claim 9, wherein temporally adjacent applications comprise at least two applications in sequential time order.