Analytics for the Internet of Things

Mathias Funk
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands
m.funk@tue.nl

Piet van der Putten
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands
p.h.a.v.d.putten@tue.nl

Henk Corporaal
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands
h.corporaal@tue.nl

CHI 2009, April 4 – 9, 2009, Boston, Massachusetts, USA.
ACM 978-1-60558-247-4/09/04.

Abstract
This paper presents ongoing work on an approach to remotely observe the usage of connected products, analyze collected data and dynamically refine the observation mechanisms for better data. This allows for iteratively working towards the most elaborate, meaningful, and relevant representation of usage behavior in the form of structured and semantically annotated data traces. We show an implementation of the approach in the D'PUIS framework.

Keywords
User experience, evaluation, remote observation, Internet of Things, visual language, methodology

ACM Classification Keywords
H.5.2 User Interfaces: *Evaluation/methodology*

Introduction
Nowadays, the almost ubiquitous Internet is accessed not only by traditional computing machinery, but also by new types of devices, connected products. These products offer a certain, often specialized functionality to users, enabled by a world-wide connection to several online services. The devices that form the Internet of Things [5] are diverse and may range from RFID tag equipped packaging material to complex electronic products. The latter aim at entertainment, mobile applications, or productivity, and specialize on tasks

that could be carried out easily by personal computers, but are placed into new contexts, realize a better user experience, or package functionality aspects in a new, cheaper, or more robust way. Since these connected devices implement a strong relationship to the usage context, evaluation has to be carried out under similar conditions, data acquisition in the user's habitual environment becomes crucial.

However, at the same time, methods for doing such an evaluation must evolve, too. Due to high market pressure and shorter design and development cycles, user experience testing which is done in the traditional way seems to be too slow to be acceptable in terms of feasibility and effort to designers, engineers, quality experts, in short, domain experts, in charge.

The Internet, being the technical basis for such devices and related business cases, can also serve as a setting that enables cheap and light-weight user experience evaluation services. We would like to draw an analogy to Google's Analytics service [6] (and also to other website statistics providers) that provides easy, but powerful website usage statistics to website owners – accessible to both small homepages and huge enterprise-grade web portals. This service offers a very simple integration mechanism (by means of JavaScript snippets integrated into the HTML source of the webpage), and powerful tools to leverage the acquired data. Very similarly, our approach aims at reducing the effort needed to integrate observation into products, while providing extensive tools for (creatively) working with the gained observation and analysis capabilities. The approach presented here aims at changing the evaluation process towards a more parallel and streamlined activity. This allows multiple domain experts to carry out experiments at the same time, as well as it allows then to change experiments on-the-fly to derive more refined data.

In the remainder of this paper, we first point at related work, then show the approach. This is followed by a description of our implementation. Finally, we conclude the paper and give an outlook on future steps.

Related work

The use of monitoring systems to acquire usage information from remote systems is not new [2,7,8]. Such approaches have been incorporated both for maintenance monitoring and fault-detection purposes as well as ways to collect data beneficial for evaluation tasks. The research presented here and in [3,4] differs in three main points from similar approaches: the use of a *more user-friendly* visual language for observation specification, the capability to *redefine experiments on-the-fly*, and the use of an *engineering approach* that renders the technology applicable to a wide spectrum of connected products, broadening the circle.

Approach

Iterative observation is a process divided basically into four parts: (1) conceptualization, (2) observation definition, (3) automatic data collection, and (4) data analysis. With a high degree of automation in step 3, human capabilities and experience can focus on essentially *human* tasks, such as conceptualization, definition and dissemination. Figure 1 shows the connection of these process parts.

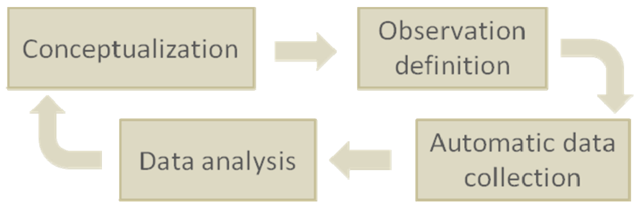


figure 1. Iterative approach to collect meaningful and relevant data with the D'PUIS framework.

In the *conceptualization* phase, domain experts think about the semantic structure of the data to be collected. Often this involves also semantically structuring the product's functionality or non-functional aspects such as user experience, fault-tolerance, or performance characteristics. This knowledge should be formalized in terms of ontologies (graphs of semantic concepts), to be usable as input for the observation framework.

During the *observation definition* phase, more or less vague *questions* about user behavior are cast into more elaborate *metrics*, e.g. elaborating "How often is function A used?" to "What is the average number of function_A_used triggers per day per user?" Similar to the first phase, this informal mapping task results in a formal specification of observation which is a direct input for the observation framework. In this phase, the data collection is parameterized on an abstract level using a simple visual language. This language allows for a graphical connection between data sources, processing nodes, semantic nodes and data handling blocks, intuitively visualizing a flow of information.

Subsequently, in the *data collection* phase, observation specifications are automatically distributed to all connected products (that "participate" in the experiment), and observation modules within the products start to collect data according to the specifications. The data items are fed back to a central data collection unit (cf. Figure 3, "server component") and remain in a storage space.

While the data is collected, the central data collection unit provides a real-time view of aggregated data and also allows for visualizations of incoming data. This real-time visualization can often answer first questions such as, whether the right and relevant data is being collected. In addition, it helps to highlight interesting parts the overall data space that required deeper digging and, thus, a change of observation specifications.

However, the closing of the cycle is crucial. Collected data can be accessed and analyzed at any point in time during the experiment, and, consequently, the data collection process itself can be adapted dynamically anytime. Moreover, different data collection mechanisms can work in parallel, each e.g. used by a different domain expert to acquire a dedicated view on product usage. This approach presents the experimenter (often also a group of experimenters) with a way to maximize the use of costly experiment time and participants.

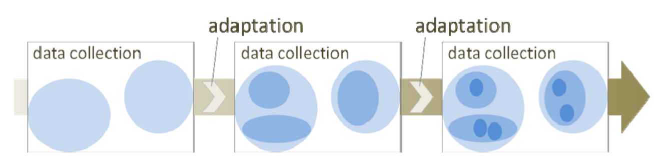
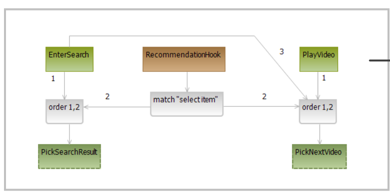


figure 2. Iterative deepening of data and understanding.

Figure 2 visualizes the outcome of such an approach: during a first – explorative – data collection phase, two vaguely interesting areas of information can be identified, after a short adaptation of observation, new data is collected that concentrates on the two areas, yielding three even more refined areas, and finally, after another iteration, five reasonably precise sets of data items appear that represent the sought-after data.

Implementation

The approach has been implemented with the Dynamic Product Usage Information System (D'PUIS) which includes and connects an authoring environment, a server component, and an observation module that is integrated into the product(s) under observation. These three high-level components will be explained below.



Screenshot of the authoring environment showing the visual language editor for observation specifications

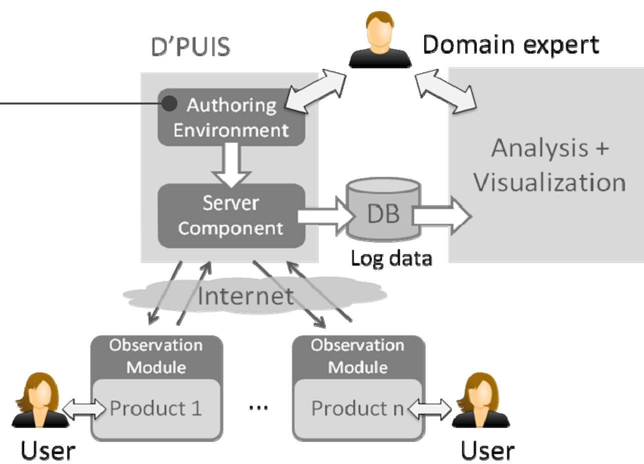


figure 3. Technical system overview.

The *authoring environment* helps domain experts in creating observation specifications and surveys that instruct observation modules to collect certain information from the products. Therefore, the authoring environment provides visual and textual editors to achieve this easily.

Observation modules are integrated into the products in such a way that they have access to product information that represents user behavior, system event sources are abstracted and observation specifications (created in the authoring environment) define how information coming from these sources is collected and even pre-processed. Due to the tight integration of observation modules into the host products, performance limitations, as well as security and privacy constraints have to be addressed. In the current state of development, platforms such as .NET and JAVA are supported. However, we are now working on observation modules for embedded devices.

Both authoring environment and observation modules are connected via a *server component*. This component first manages all active observation tasks, and, second, aggregates the collected data. It maintains the link to all connected observation modules, and keeps statistics about the availability of experiment units; that is, whether a system is online and collecting data. In terms of storage, the server component uses a database which can be used for an instant visualization of acquired information and data export to 3rd party tools, such as dedicated analysis tools like ProM [1] that tackle special problems.

The realization of the approach aims at abstracting from purely technical tasks such as distribution of specifications to connected devices, routing of data items, preprocessing, and performance issues in observed products. This enables domain experts to (1) concentrate on the information matter, and (2) to easily instruct a potentially large network of distributed products, potentially stretching over several cities, countries or even continents[1]. The framework serves as a virtual connection between the user and their behavior, and the domain expert, although they might reside in very different places during the experiment.

As an example, consider a home-entertainment product that supports browsing and searching of video content several ways, incorporating different usage paradigms. The implementation of our approach, D'PUIS, allows for instrumenting these devices with observation modules, and distributing the products to key testers' homes. Now, the detailed usage of the system within a natural usage context such as a living room can be observed and explored remotely in a dynamic way.

Conclusions and Outlook

The domain of connected products is soaring and will sooner or later form a significant share of the Internet of Things, a world-wide network of interconnected physical devices. This mass of different products is a challenge for user experience professionals, both in academia and in industry.

We presented an approach to leverage the connectivity, and use the Internet for an observation framework. The framework, D'PUIS, enables domain experts to visually define data collection tasks and start analyzing acquired data, while the task of actual data collection is left to the framework. The framework allows for a flexible change of data collection anytime and, thus, enables a new way of getting insight into user behavior. The use of the Internet connectivity allows for placing experiment units in a natural usage environment, and provides remote access to usage data transparently. However, this approach to user experience research poses different biases than traditional lab-based testing. Furthermore, it has serious implications on the user's privacy which have to be addressed carefully. This involves opt-in and opt-out mechanisms and transparent communication of observed data to the user. However, future directions are to involve users more prominently in the creation of appealing products.

We aim at providing a technical basis for experiments that are concerned with both the emerging field of connected products, and that is at the same time applicable to other user experience evaluation approaches. Coming back to the analogy of website analytics and our approach, the framework presented is

- *inexpensive* and *accessible*, since it is web-based and built on *open* frameworks and a number of platforms,
- *iterative* and *dynamic* in terms of data collection and analysis capabilities,
- *easy to use* and *implement*, since the programming effort is minimized and a large share of tasks are automated
- *scaling* due to the high degree of automation and capabilities to connect a high number of products to the server component.

[1] We evaluated D'PUIS in several case-studies, in one study distributing observed products to 8 countries world-wide, including the US, Singapore and European countries.

While the framework has been implemented and tested in case-studies, all being in the context of prototype testing, future challenges remain for the integration of observation functionality into released products. Likewise, developing methods that are enabled by these iterative testing ideas have still to be explored. The data collection mechanisms shown in this paper concentrate on quantitative data, omitting “subjective” data collection mechanisms like surveys and feedback forms which are built into the D’PUIS framework, but beyond the scope of this paper. However, we imagine that new technical approaches towards evaluation of novel connected products are indeed necessary for the success of emerging products and services, since users become more concerned about “soft” product failures that can be attributed, e.g. to poor usability, and will consequently reject failing products and brands. We see another positive aspect of our approach in increasing acceptance of user experience testing, especially in industry. Even today, a great reluctance towards extensive user testing of products can be found. Our approach might ease the transition to good testing practices and, thus, better products.

Acknowledgements

This work is being carried out as part of the “Managing Soft-Reliability in Strongly Innovative Product Creation Processes” project (www.softreliability.org), sponsored by the Dutch Ministry of Economic Affairs under the IOP-IPCR program.

References

[1] W.M.P. van der Aalst, B.F. van Dongen, C.W. Günther, R.S. Mans, A.K. Alves de Medeiros, A. Rozinat, V. Rubin, M. Song, H.M.W. Verbeek, A.J.M.M. Weijters, ProM 4.0: Comprehensive Support for Real Process Analysis. Proc. ICATPN 2007, Springer-Verlag (2007), 484–494.

[2] J. Chin, V. Diehl, K. Norman, Development of an instrument measuring user satisfaction of the human-computer interface, Proc. CHI 1988, ACM Press (1988), 213–218.

[3] M. Funk, P.H.A. van der Putten, H. Corporaal, Model interpretation for executable observation specifications. Proc. SEKE 2008, 785–790.

[4] M. Funk, A. Rozinat, A.K. Alves de Medeiros, P.H.A. van der Putten, H. Corporaal, W.M.P. van der Aalst, Improving product usage monitoring and analysis with semantic concepts, Proc. ISTA 2009, to be published.

[5] N. Gershenfeld, R. Krikorian, D. Cohen, The Internet of Things, Scientific American, October 2004, pp 76-81.

[6] Google Analytics, http://www.google.com/analytics

[7] D.M. Hilbert, D.F. Redmiles, An approach to large-scale collection of application usage data over the internet, Proc. ICSE 1998, 136.

[8] J.H. Kim and D.V. Gunn and E. Schuh and B. Phillips and R.J. Pagulayan and D. Wixon, Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems, Proc. CHI 2008, ACM Press (2008), 443-452.