

Linking quality of service and experience in distributed multimedia systems using PROV semantics

Stephen C. Phillips, Maxim Bashevoy, Michael Boniface,
Simon Crowle, Vegard Engen, Zipani Sinkala, Stefanie Wiegand
IT Innovation Centre,
Gamma House, Enterprise Road,
Southampton, UK
{scp, mvb, mjb, sgc, ve, zts, sw}@it-innovation.soton.ac.uk

Abstract—Experimenters creating innovative applications that combine diverse distributed multimedia services with rich end user applications require enhanced insight into the relationships between the perceived quality of experience (QoE) and provided quality of service (QoS). We have implemented software which not only captures QoE and QoS measurements but, using a provenance ontology, also records the interactions between end users, the content, applications and the services. The data exploration interface provided allows an experimenter, working with participants in real-world situations, to understand the detail of the participants' usage and experience of the system and the system performance factors contributing to their quality of experience.

Keywords—provenance; PROV; SOC; QoE; QoS; UX; HCI; EXPERImonitor;

I. INTRODUCTION

New media applications and services are revolutionizing social interaction and user experience in both society and in wide ranging industry sectors. The rapid emergence of pervasive human and environment sensing technologies, novel immersive presentation devices and high performance, globally connected network and cloud infrastructures is generating huge opportunities for application providers, service provider and content providers.

These new applications are driving convergence across devices, clouds, networks and services, and the merging of industries, technology and society. Yet the developers of such systems face many challenges in understanding how to optimize the Quality of Service (QoS) of their solutions to enhance the users' Quality of Experience (QoE) and how their disruptive innovations can be introduced into the market with appropriate business models.

The EXPERIMEDIA project [1][2] brought together innovators from academia and industry to build a platform for experimenting with these distributed multimedia technologies

in three diverse venues: a ski resort, an athletic training facility and a museum. The EXPERImonitor software described here was fundamental to understanding the 20 experiments performed in the project.

EXPERImonitor [3] is a framework focused on the management of experiment content that allows experimenters to explore the relationship between QoS and QoE in complex distributed multimedia systems. The tool is specifically designed to support the observation of systems where user-centricity, mobility, ad hoc participation and real-time access to information are critical to success.

EXPERImonitor uses a hybrid data model that combines formal, often numeric, metric reporting with semantic provenance information described using W3C PROV [4]. The hybrid approach provides the ability to collect large quantities of measurement data (e.g. service response times, network latency, user satisfaction, etc.) whilst allowing for exploration of causation between observations within such data (e.g. user satisfaction in relation to service response time).

The ability to efficiently traverse experiment content between QoS and QoE is an essential capability for evaluation of complex socio-technical systems. Data exploration can provide indications of factors that influence each other and is used to segment data for further investigation and analysis. With ever growing big data sets generated by Internet systems, EXPERImonitor can significantly reduce the time from observation to insight.

In the remainder of this paper we first discuss prior work in the fields of measuring quality of experience and of service and of using provenance data to understand workflows in SOC. In Section III we present scenario that motivates our work in understanding user interactions to link QoS and QoE. In Section IV the theory of the data models we have developed is described and then in Section V we describe the implementation of the models in EXPERImonitor and return to the motivating scenario to illustrate its utility. The paper concludes with thoughts on future work.

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287966.

II. BACKGROUND

In this section we describe in more detail the concepts of Quality of Service, Quality of Experience and of Provenance. We describe the background work that we have built upon and how these concepts relate to our work.

A. *Quality of Experience and of Service*

Research investigating the relationships between user experience and interactive technologies is in a relatively early stage of development; currently there is no single, sufficiently well-developed theory that provides a framework upon which our work in EXPERIMEDIA can be directly based. We approach the understanding of QoE from the context of experiments undertaken in real-world, not laboratory, environments. Rather than taking the narrow point of view of QoE being an extension of network QoS and just measuring the performance of delivered services from an end-user perspective [5] we take the broader view of QoE espoused in [6]. In [6] QoE is taken to be influenced by the user's context (e.g. motivation, social context, personal situation), their experience (e.g. prior experience, expectations) and the QoS of the system (e.g. performance, availability).

Research has shown that appeal of a software system comes from ergonomic factors (e.g. simplicity, controllability) and hedonic qualities (e.g. novelty, originality) and these personal perceptions and experiences of a user's interactions with a product or service play a significant role in their acceptance and use of the technology in the future [7]. However, capturing and understanding the experiential dimension of a human-computer interaction is a methodologically challenging goal. Unlike other, externally and objectively measurable quantities (such as the user's time taken to locate a specific e-mail item), experiential measures are inherently subjective and only indirectly recordable. Therefore some care must be taken to collect this data using a foundation and framework that most effectively communicates meaningful results to experimenters that want to better understand the impact of their technology on end users and how the technology or product can be improved.

An individual's active and remembered experiences with computing technologies are in part shaped by the interactions he or she has with the technology in the context of the environment and their personal needs and goals [8]. This ties with Norman's theory of action [9] from the human-computer interaction research community:

- 1) Create/revise a desirable goal or outcome
- 2) Formulate/revise a plan to achieve a goal
- 3) Select an action to enact to advance towards the goal state
- 4) Physically enact the action
- 5) Observe the response of the environment to the action
- 6) Interpret the response in terms of the expected outcome
- 7) Evaluate progress towards the goal state

Potentially deleterious impacts on the user both from the point of view of objective measures of usability and also qualitative indicators of experience can occur where there are

mismatches between the user's goal and intentions and the environment's attributes or affordances made available to address them.

The primary method used to capture attitudinal QoE measurements in the EXPERIMEDIA experiments was the use of questionnaires using the visual analogue scaling method devised by Rensis Likert [10]. Here, a series of items that relate to an experimental component (such as "ease of use": "The dashboard easy to understand") are presented to the respondent along with a horizontally marked scale positioning agreement from negative to positive (typically a 5 or 7 point scale with a middle, neutral value is used). It is not unusual for a Likert scale to address multiple components within a randomized statement set. Data collected using this method, provided it can be safely assumed to come from a normal population, can be analyzed using a number of common correlation and factorial methods. Likert's scale is a relatively quick and simple way of gathering positive and negative attitudinal responses. However, some data sets can be effected by various biasing factors, particularly when users are asked to rate a statement that reflects directly on themselves an organization related to them.

Quantitative measurements of a participant's experience with a computer system such as those collected through a Likert scale are recorded in our system. In addition, a novel aspect of this work is the capture of user interactions with the application (a type of QoE) and with the rest of the system using a semantic model (see below).

The experimenter is not only interested in the QoE of the participants but in the potential causes of the QoE, whether good or bad. Although many factors affecting in particular a participant's attitudinal self-reporting cannot be known or measured, the performance of the system the participants interact with can be measured and this is what we term Quality of Service. A simple example of QoS affecting QoE would be the response time of a web-service compared to a participant's reported enjoyment of using the service, where a long response time would be expected to decrease the reported enjoyment.

Linking QoE to QoS has been done in some experiments in controlled environments such as understanding the perceived quality of multimedia environments in a laboratory setting [11] or user evaluation of a web site [12]. In both these cases the wide range of factors discussed above that can influence a user's QoE were controlled as much as possible and the authors were able to perform a mathematical analysis of the QoS/QoE correlations. The EXPERImonitor system is designed for real-world settings where many factors are uncontrolled. EXPERImonitor does not try to compete with statistical packages but instead can export the recorded data for statistical analysis in other tools. Natively, an exploration interface is provided to help the experimenter understand the events that have occurred and the relationships between factors.

B. *Provenance*

The W3C Provenance Incubator Group defined provenance to be "a record that describes the people, institutions, entities and activities involved in producing, influencing, or delivering a piece of data or a thing". In EXPERIMEDIA's distributed multimedia experiments, the content acquired by and delivered

to the experiment participants is of primary importance and by understanding the provenance of the content (how it was created, delivered and consumed) we can understand the most important experiment interactions.

The recently standardized W3C PROV Family of Documents [4] defines a model, corresponding serializations and other supporting definitions to enable the inter-operable interchange of provenance information in heterogeneous environments such as the Web. In this work we have used the PROV-O ontology: an OWL2 ontology defining how to map the PROV data model to RDF.

PROV-O defines three fundamental notions:

- Agents: cause changes to entities in the world through *Activities*. *Agents* might be human actors, organisations or pieces of software acting on someone’s behalf. *Agents* can also be *Entities*.
- Activities: describe how *Entities* are created, used and changed. They cannot be *Entities*. *Activities* may have associated times at which they are “started” and “ended”.
- Entities: the resources whose provenance we wish to describe. *Entities* may have associated times at which they are “generated” or “invalidated”.

These three classes along with a rich model for linking defining usage, association, generation, derivation, etc, provide a scheme for recording “who did what to what when”.

For our work we have used PROV-O to record interactions between the experiment participants and the software system and between services composing the software system. This is in many ways similar to recording a distributed scientific workflow using provenance as implemented in the Taverna

workflow management system [13] for instance based on previous work in myGrid [14]. The Taverna system uses the central workflow enactor to record provenance data but there has also been work done on distributed systems recording the provenance information centrally [15]. This is the concept used in EXPERImonitor but taking provenance data from a wide variety of clients rather than just grid services.

In the work of [16], an extension of W3C PROV is defined which, when combined with a service level agreement (SLA) ontology [17], records the SLA, service performance monitoring data, QoE feedback and assesses the credibility of the feedback by recording its provenance. In contrast to this, our work has been implemented and used in real experiments, records QoE and QoS data in a high-performance relational database and uses provenance to record system interactions and thus link one dataset to the other.

III. MOTIVATING SCENARIO

The “Smart Ski Goggles” experiment from the EXPERIMEDIA project [18] provides a clear motivating scenario for gathering user-interaction data and using it to help understand the links between QoE and QoS data. In this experiment, tourists at the Schladming ski resort were asked to wear a new design of ski goggle containing a small screen that could display information to the skier by linking to a novel application running on the participant’s smart phone. The application is controlled via a simple set of buttons on a wristband so that it can be operated whilst still wearing gloves and leaving the phone in the pocket.

Fig. 1 shows a simplified deployment diagram of the experiment along with the primary data links. During the day’s skiing each experiment participant (the “Skier”) may choose to use various features of the application (via the wristband and

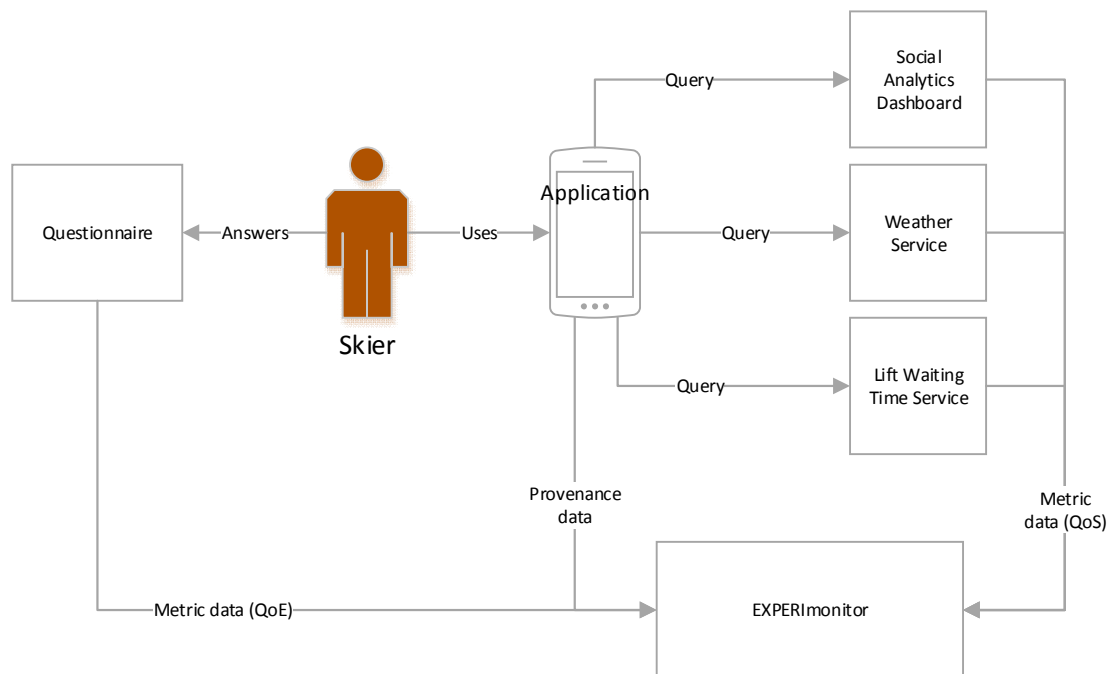


Fig. 1. Deployment diagram for the motivating scenario showing how each skier (indirectly) uses a smart-phone application to interact with three information services. The data from the questionnaire, from application interactions and from each information service is sent to the EXPERImonitor service.

the goggles) which in turn query various information services behind the scenes. The three services shown here are the “Social analytics Dashboard” which provides the most relevant tweets for the region, the “Weather Service” for short-range forecasts and the “Lift Waiting Time Service” which indicates how long the waiting time is likely to be at each of the nearby ski-lifts and therefore potentially influencing the skier’s choice of route down the mountain. At the end of the day, each participant is asked to fill in a questionnaire about their experience of using the application. How the application works and the services it uses are not known to the participants: it is just their direct experience of the application that is requested.

When reviewing the data from the experiment participants, the experimenter would be interested to understand the answers to questions such as:

- Who used which application features, how often and when?
- If a participant was dissatisfied in any way, what was the cause?
- Did participants’ (dis)satisfaction correlate in any way to the performance of the information services?

To answer these questions we need to:

- record the QoE of the participants: both the QoE reported through the questionnaire and the interactions with the application;
- record the application’s interactions (on behalf of the participant) with the information services;
- record the QoS of the information services;
- analyze and present the data in such a way that the experimenter, with their additional knowledge of the experimental scenario, can explore the data and tease out the answers.

The Method and Implementation sections below describe how we have created a system for experimenters to do just this.

IV. METHOD

To capture the necessary data we define two linked data models to be used by EXPERImonitor clients sending data to the central data store:

- 1) a metric model to record entities, attributes and their measurements, and
- 2) an extension to the W3C PROV ontology and a set of interaction patterns to record the interactions between different actors (also entities) in the system.

A. Metric Model

The metric model is shown in Fig. 2. It describes the entities, their attributes and measurements of those attributes along with logical groupings and meta-data to aid analysis.

In an *Experiment* a *MetricGenerator* (instantiated in a piece of client software) creates many *MeasurementSets* (logically grouped into *MetricGroups*) with each *MeasurementSet*

recording multiple *Measurements* (along with the time of the measurement) of a single *Attribute* of an *Entity*. A *Measurement* is described using a *Metric* which has a *MetricType* (using the Nominal, Ordinal, Interval, Ratio typology) and a *Unit*.

The complexity of the model enables it to support the situation where there are multiple observers of an entity’s attribute in a single experiment, potentially using different metrics. For instance, a light sensor attached to a computer and reporting the primary color of an object using its wavelength (a “Ratio” such as 475nm) and a human reporting qualitatively that the object is “blue” (a piece of “Nominal” data).

In our scenario, the QoS data from the information services (such as their average response time, and momentary CPU and memory load) are recorded as metric data with each service being a separate *Entity* with separate *MetricGenerators* but using common *Metrics*. The QoE data from the questionnaires is also recorded as metric data using this model. In this case the *Entity* is the participant, the *MetricGenerator* corresponds to the online survey tool, the *Attributes* are the questions and the

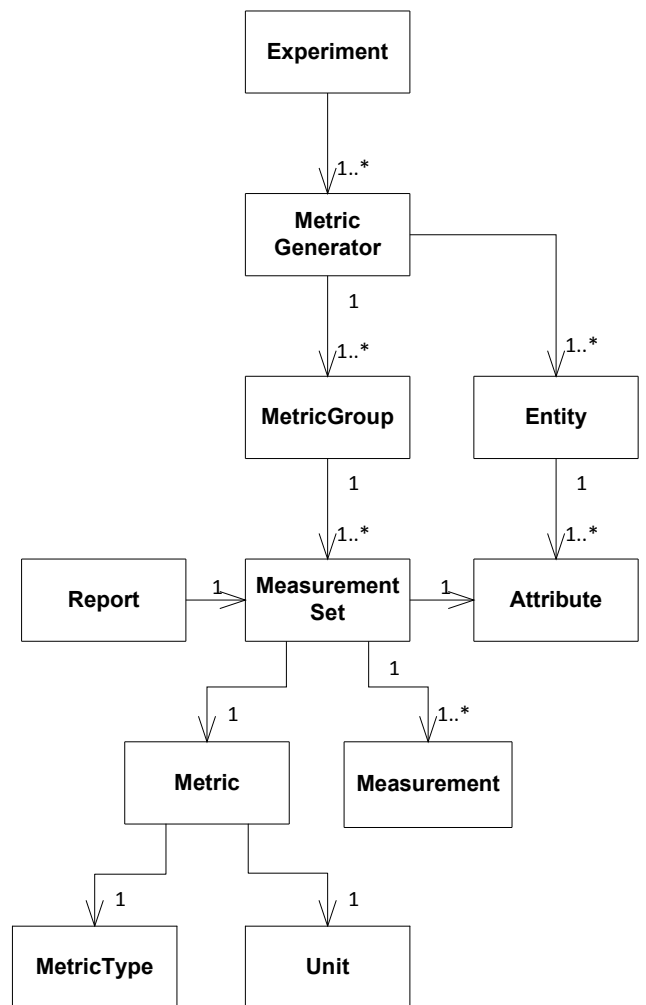


Fig. 2. The metric model for EXPERImonitor. Within an Experiment Each monitoring client has a MetricGenerator and reports Measurements of Metrics but the Entity and Attribute model may be shared amongst multiple clients.

Measurement the answers. The *Metric* definition includes additional meta-data to inform the system that it is describing Likert data.

B. Provenance Model

To capture the interactions between the participant and the application and the application and the information services we have defined an extension to the W3C PROV ontology and some interaction patterns.

The EXPERImonitor provenance model is initially scoped to cover a subset of the PROV-N schema [19]:

- Component 1: Entities, Activities, Generation, Usage, Start, End, Invalidation
- Component 3: Agent, Attribution, Association

Building on these concepts, a small ontology has been created to define the key concepts in the EXPERImonitor data model: *Participant*, *Content*, *Application* and *Service*. These classes are shown in Fig. 3 and are used to help create interaction patterns and simplify subsequent queries on the data. In this and following Figures we have used the standard notation of a pentagon to represent an Agent, an ellipse for an Entity and a rectangle for an Activity. The *Application* in the model uses both an *Entity* and an *Agent* because the *Application* must be used by a *Participant* in *Activities* (and so is an *Entity*) and causes change itself acting on behalf of the *Participant* through *Activities* interacting with *Services* and so must also be an *Agent*. This duality concept could easily be

extended to *Services* to support service composition patterns.

Within the scope of the supported PROV data framework, additional semantic data may also be included to offer the experimenter further opportunities to examine behavior within domain specific ontologies. For example, where it is possible for an EXPERImonitor client to do so, it could include additional semantics taken from specific ontological domains, such as SIOC [20] or FOAF [21]. For instance, the *Participant* uses the foaf:Person class and is ready for additional FOAF annotations. If this additional data can be provided by clients then it can enhance power of the queries performed on the data.

The W3C PROV model is complex and provides considerable freedom of expression. This is both an advantage and a drawback: it is able to express useful information in a wide variety of situations and domains, but automated analysis of the data is difficult unless additional structure is imposed. With this in mind the EXPERImonitor architecture recommends a set of interaction patterns (encoded in the client API) be used to record interactions between the classes defined above. These patterns currently cover the common cases of *Participants* interacting with *Applications* (on phones, tablets or larger computers) and those *Applications* interacting with *Services*. Through these interactions, *Content* is created and consumed and this is also modelled.

Patterns have been defined for a *Participant* using an *Application*, a *Participant* creating *Content* on the client (for instance taking a photo), *Participant* creating *Content* at a *Service* (uploading a photo) and *Participant* retrieving *Content* from a *Service* (downloading some data). In Fig. 4 and Fig. 5 the PROV classes are just represented by the standard element shapes and other annotations (included in Fig. 3) are also omitted for simplicity.

C. Linking the Models

We have described two models for capturing data during an experiment: a metric model used for measurements of both participants' reported QoE and services measured QoS and a provenance model used for recording interactions between participants, applications and services in relation to content. The metric model on its own does not encode any relationships between the entities. The provenance model does link entities via their interactions and therefore provides the link between the QoE and QoS data held in the metric model.

As an example, let us consider how a QoE metric representing a questionnaire answer can be linked to the QoS reports of a service. First, the measurement of the QoE metric is linked through the metric model to a participant entity. The participant entity in the metric model can be linked to the *Participant* in the provenance model and via the reported provenance interaction patterns to a *Service* that the *Participant* used (via the *Application*). The *Service* in the provenance model is linked to the service entity back in the metric model and then to the time series data measuring the QoS performance for that service.

By linking these models we can understand who reported what QoE and what services they used. The model does not specifically link individual QoE reports to individual services

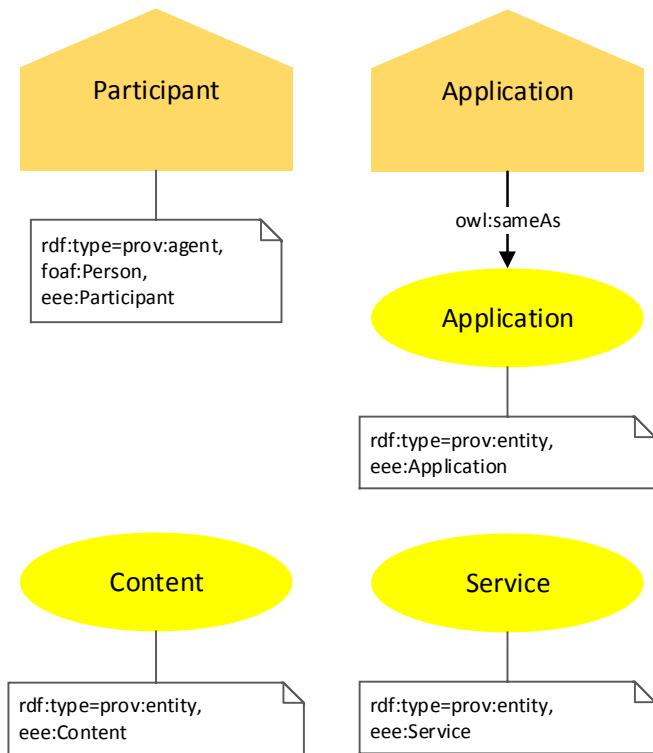


Fig. 3. Participant, Content, Application and Service are key concepts for EXPERImonitor and are each represented by small additions to the standard PROV-O classes using an rdf:type with the “eee” or “http://experimedia.eu/ontologies/ExperimediaExperimentExplorer#” namespace

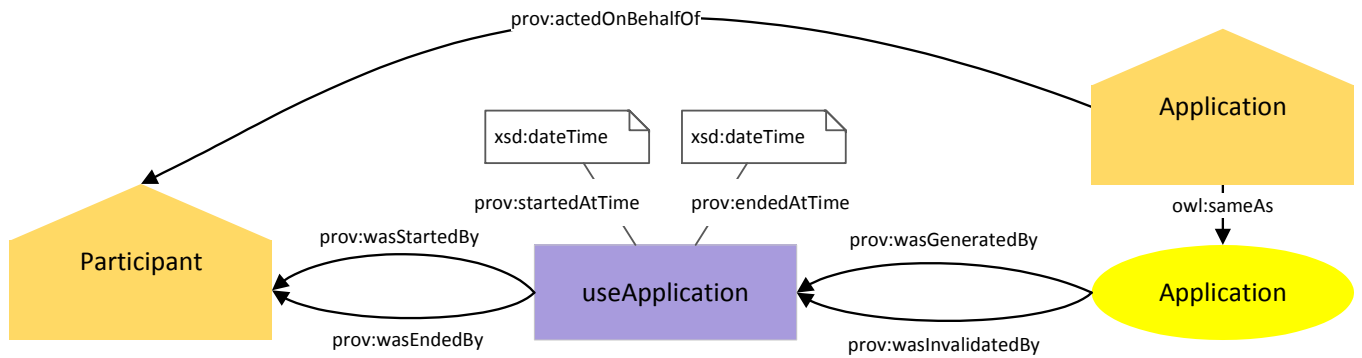


Fig. 4. Interaction pattern for when an experiment participant uses an application. The application here represents an executing instance of some stored software and the activity start and end time correspond to the start and end times of the application usage.

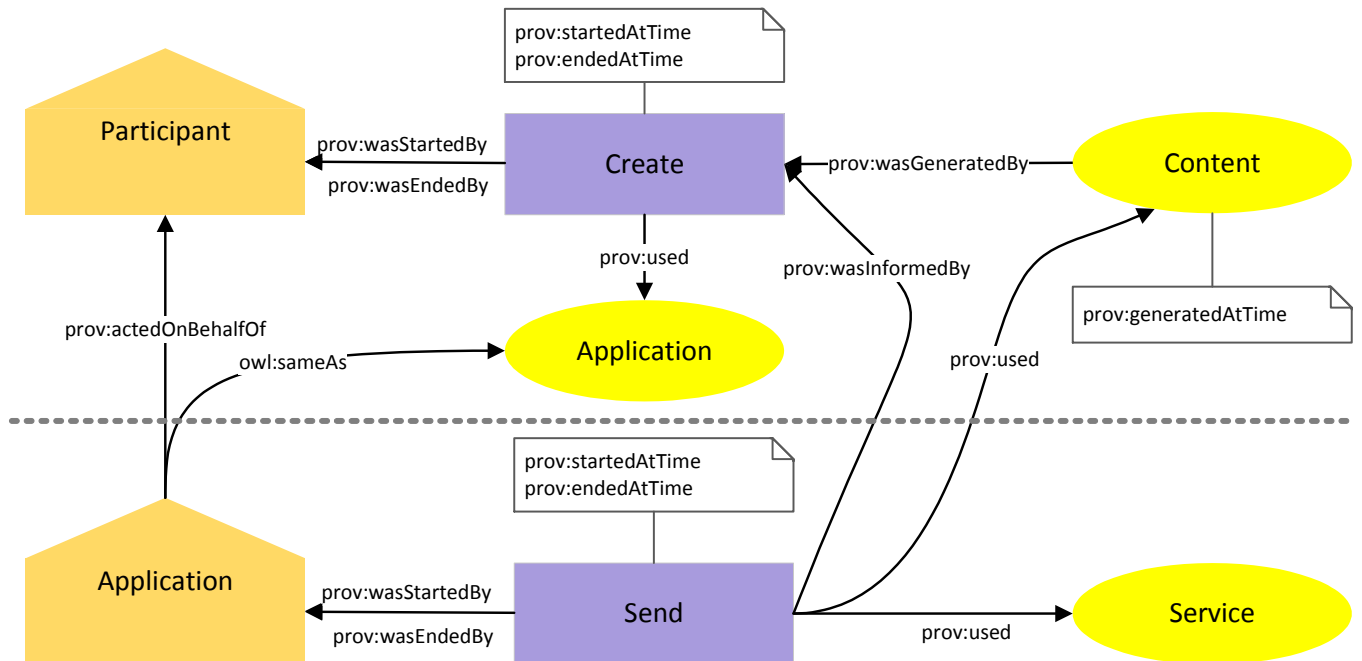


Fig. 5. Interaction pattern for an experiment participant uploads some content to a service. Here the content is created by the participant using the application and then the application acts for the participant in sending the content to the service. The upper half of the figure is also the pattern for creating content locally on a device (such as taking a photo).

as we recognize that QoE reports are generally not that simple and encompass many aspects of a participant's experience (see above). The experimenter may infer additional information based on their understanding of the experimental context, the times and potentially locations of the *Activities*. An example of this provided in the Implementation section below.

Data gathered in experiments can come from many different sources, all instrumented to report measurements using these two models: for instance hosted services provided by other parties, or services and mobile clients deployed by the experimenter. To make sense of the data and enable the linking just described, where possible the different independently developed data sources (EXPERImonitor clients) must use the same identifiers for the same entities. For instance, a hosted service (entity) may report QoS measurements about itself and this data needs to be linked to reports from an experimenter's client application that uses the service.

Here we define the principles for naming entities:

- Entity identifiers are assigned when an entity is born by another entity responsible for creating them.
- Entity identifiers should be unique enough so that they do not clash in a context of use (e.g. within a set of experiments).
- Entity identifiers should be structured according to URIs where possible.
- Entity identifiers based on URIs can be dereferenceable but this is not mandatory.

In the metric model, the entity has four fields: a UUID, Name, Description and entityID. The entityID field must be set to be the standard identifier for that entity and this must also be used as the subject URI for the related class instance in the provenance model. For instance, a service reporting its QoS would use its own URL as the entityID and the URL would

also be used as the URI in the Service instance by any interacting agents.

V. IMPLEMENTATION

The metric and provenance models described above have been implemented in the open source EXPERImonitor software. EXPERImonitor is a Java web service using the Spring framework with a web based experimenter's interface written using Foundation, JQuery and the D3.js graphics library. The EXPERImonitor software is supported by several external services and databases:

- a WebDAV configuration service and store which maintains shared configuration data for all the services and clients (Apache HTTP Server);
- a triple-store for the provenance data (OWLim);
- a relational database for the metric data (PostgreSQL); and
- an AMQP bus to connect to the monitoring clients (RabbitMQ).

The EXPERImonitor software comes complete with client APIs for development in Java (also on Android), C# and C++.

Returning to our motivating scenario, the Java, Android and C++ clients were all used in the Smart Ski Goggles experiment. The data generated by the Smart Ski goggle experiment was analyzed and reformed to create the simulated data presented here in order to more clearly illustrate the concepts we are discussing.

At the end of a day's experiment with 10 skiers, the experimenter can review the QoE data in the EXPERImonitor web dashboard via the "QoE widget" which queries first for all Participants (from the provenance database) and then for all QoE Likert data linked to those same participant entities in the metric database. The data can be presented both in summary form and in detail for an individual question (Fig. 6). Clicking on elements of the QoE charts lists the Participants who reported a particular QoE response and provides links to open further widgets. In this case we imagine that the experimenter, concerned about the poor "responsiveness" report has clicked on that and goes on to explore the activities of "Bob".

The activities widget is a representation of the interactions of an individual Participant. Using the interaction patterns discussed above, the provenance database can be queried to discover all the Activities undertaken by a Participant which used a Service. We present the information simply as a table with sortable columns: Activity, Application, Service, start time, end time and duration. In our scenario, we see that Bob tried various features of the application early in the day and did experience a long response time when using the lift waiting time service on the three occasions he tried to use it. We would also see that Bob did not use the application at all after 09:59 and, as an experimenter using knowledge of the experiment context (and even of Bob) might know that Bob did continue skiing after that time and suppose that Bob got fed up with the application because of the poor response time and gave up on using it.

Alerted to the poor response time of the lift waiting time service the experimenter is able to click through to the final widget which presents time-series QoS data from services. This QoS widget (Fig. 7) does more than a standard service performance chart as it is able to overlay the times that any of the Participants performed any Activities. A reasonable overlay in this case (which is opened automatically from the previous widget) is that of overlaying the times that Bob checked the lift waiting times on top of the performance measures for the lift waiting time service (and in particular its response time). As has been previously noted, in complex distributed systems, such direct causes of poor QoE often do not exist and certainly cannot be inferred from the data even in as rich a model as this. For this reason, the interface provides the ability to overlay any Participant's Activities on any QoS metric in case such an overlay can provide insight to the experimenter when combined with their understanding.

By plotting the access times of the various users over the time-series data for the lift waiting time average response time the experimenter sees that the users who reported a bad QoE were unlucky enough to use the lift waiting time service at a time when it was not functioning properly and requests were timing out. Our experimenter concludes that the majority found the app useful but that the response time from the linked services is important and affects the QoE.

The reverse path may also be taken. Using summary statistics, the experimenter may spot that the response time for the lift waiting time service had a large standard deviation or range and therefore begin by investigating this aspect. Again, by using the provenance relationships, the usage times of all participants of the lift waiting time service can be overlaid (Fig. 8) and the experimenter might then segment the QoE data for the participants according to whether they used the service in the non-functioning time or not. In our example, the experimenter would then find that those participants who used the service at the non-functioning times reported a worse QoE than the ones who did not.

VI. CONCLUSION

Understanding causation in distributed multimedia systems involving experimental technologies deployed in the real-world rather than laboratories is key to learning how to improve those technologies whether in terms of user interface design, features or in performance.

We have presented a rich linked model for measuring QoS, QoE and interactions between people, applications, content and services. The model has been implemented in the open source EXPERImonitor software which uses both a relational database for high speed access to large QoS and QoE datasets and a triple-store using the W3C PROV ontology for storing and querying the interaction data.

The web interface provided by EXPERImonitor allows the experimenter to explore the data, graphically presenting information that combines data from both of the databases and helps the experimenter understand links between the reported QoE of the experiment participants and the QoS performance of the services used.

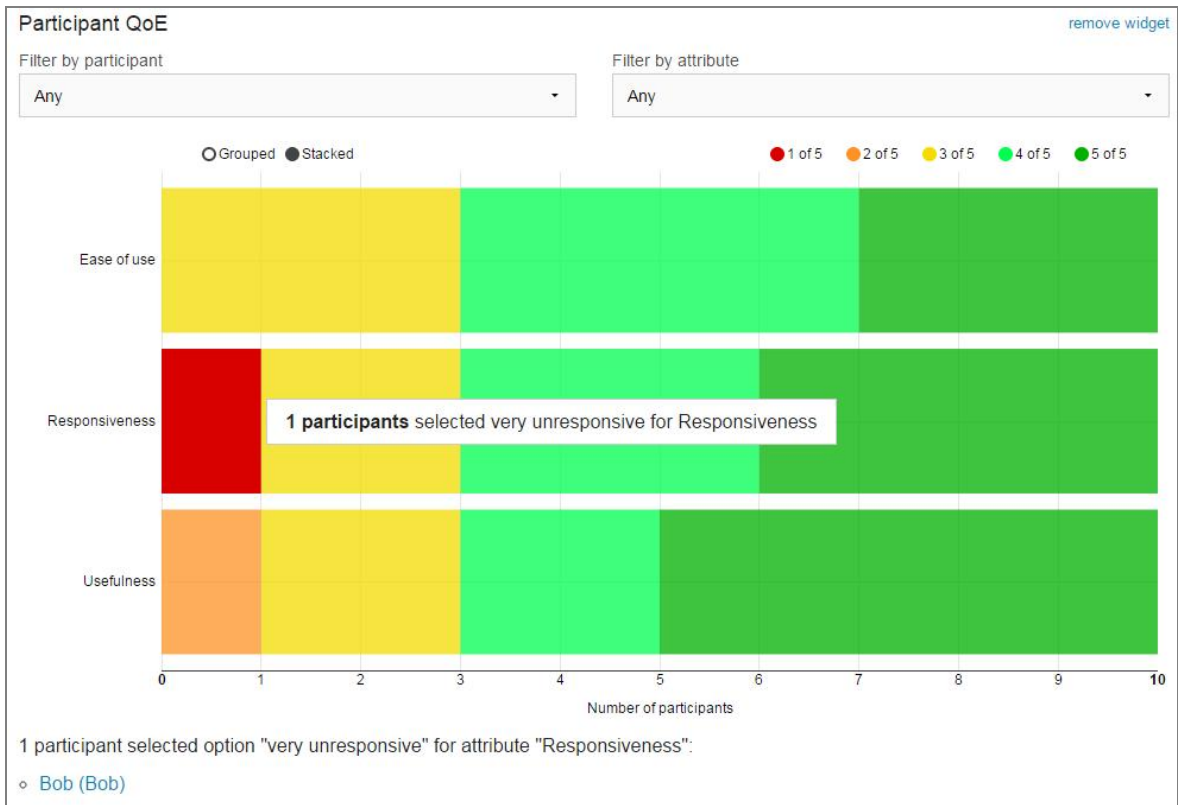


Fig. 6. QoE widget from the EXPERImonitor dashboard summarising the Likert data reported by the 10 participants regarding their use of the application. Individual participants' responses can be viewed or summary views for individual questions (attributes). Hovering and clicking on the coloured areas (the red one here) provide further information and opportunities to explore facets of the data.



Fig. 7. QoS widget from the EXPERImonitor dashboard overlaying the times that Bob tried to use the lift waiting time facility on his application on to of the self-reported average response times from the lift waiting time service.



Fig. 8. The QoS widget showing all usage times of participants who used the service overlaid on top of a zoomed in segment of the service's QoS performance data.

VII. FURTHER WORK

By extending the scope of the data collection, data models and visualizations, more insight can be provided to the experimenter. For instance, additional self-reported QoE measurements can be collected during an experiment by way of pop-up questions in the experimental application. These momentary QoE reports can add considerable value over end-of-experiment questionnaires as they capture the participants' feelings then are there. The model already records QoE data with time-stamps but the QoE visualization does not currently present the data.

Location is a second important dimension that, in some experiments, would aid understanding of the data generated by the participants. The model will be extended to store the location of the participant activities.

More visualizations of the data will aid the experimenter. The D3.js library used in our web application can support many dynamic and expressive visualizations such as parallel sets [22] and the Sankey diagram already used for PROV graphs [23]. The addition of location data will also open up possibilities for overlaying monitoring data on maps.

We intend to take advantage of semantic reasoning techniques to enrich the data that is collected in an experiment. Use of the interaction patterns affords an opportunity for automatically adding some simple additional data to the dataset but a facility for adding in experiment-specific reasoning rules would also be appropriate.

EXPERImonitor has been designed to support observation and analysis of a wide variety of systems and user experience. Future work within the ProsocialLearn project will extend EXPERImonitor to explore deeper instrumentation of cognitive factors, user affect and, increasingly, real-time analysis. ProsocialLearn will develop serious games designed to help children acquire prosocial skills necessary for positive relationships, team working, trustworthiness and emotional intelligence. In such scenarios, student outcomes must be linked to prosocial learning objectives to communicate expectations to students and to provide the basis for evaluating teacher, student, and game effectiveness. Prosocial skills must be measured efficiently and cost effectively using a series of information cues. Multi-modal signals related to game play behavior (prosocial, aggressive and socially isolated behaviors), signals hidden in game mechanics (e.g. emotion, trust, engagement, empathy) will be linked to prosociality. Vision-based facial and motion analysis (e.g. emotion,

arousal), and other profile data must be fused and evaluated in a context-dependent manner to provide quantitative indicators related to engagement which may be related to the game genre, interaction controls, etc. Driven by these signals, games will be personalized in order to achieve higher levels of player interest and, thus, maximize chances of achieving a learning objective. A pedagogically sound prosocial model will be developed based on the metric and provenance model of EXPERImonitor to store student data over time, allowing teachers to efficiently monitor progress and assess mid-long term learning outcomes across multiple games and game sessions.

REFERENCES

- [1] M. Boniface, S. C. Phillips, A. Voulodimos, D. Osborne, and S. Murg, "Technology Enablers for a Future Media Internet Testing Facility" in *Proc. 2013 NEM Summit*, Nantes, 2013, pp. 86-91
- [2] M. Boniface, S. Modafferi, A. Voulodimos, D. Osborne, and S. Murg. (2014) *Report on Features and Opportunities for Networked Multimedia Systems* [Online]. Available: <http://www.experimedia.eu>
- [3] *EXPERImonitor* [Online]. Available: <https://github.com/it-innovation/EXPERImonitor>
- [4] P. Groth and L. Moreau. (2013). *PROV-Overview* [Online]. Available: <http://www.w3.org/TR/prov-overview/>
- [5] "BT.500: Methodology for the subjective assessment of the quality of television pictures", ITU, 2012.
- [6] K. De Moor, I. Ketyko, W. Joseph, T. Deryckere, L. De Marez, L. Martens, and G. Verleye, "Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting." *Mobile Networks and Applications* vol. 15, no. 3, pp. 378-391, 2010.
- [7] M. Hassenzahl, A. Platz, M. Burmester, and K. Lehner, "Hedonic and ergonomic quality aspects determine a software's appeal." in *Proc. SIGCHI conference on Human factors in computing systems*, 2000, pp. 201-208.
- [8] M. Hassenzahl, S. Diefenbach, and A. Göritz, "Needs, affect, and interactive products—Facets of user experience." *Interacting with computers* vol. 22, no. 5, pp. 353-362, 2010.
- [9] D. A. Norman, "Cognitive engineering" in *User centered system design*, 1986, pp. 31-61.
- [10] R. Likert. "A technique for the measurement of attitudes." *Archives of psychology*. 1932.
- [11] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, and Z. Yang. "Quality of experience in distributed interactive multimedia environments: toward a theoretical framework." in *Proc. 17th ACM international conference on Multimedia*, 2009, pp. 481-490.
- [12] P. van Schaik, and J. Ling. "Five psychometric scales for online measurement of the quality of human-computer interaction in web sites." *International Journal of Human-Computer Interaction* vol. 18, no. 3, pp. 309-322, 2005.
- [13] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S., et al. "The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud." *Nucleic acids research*, 2013.
- [14] M. Greenwood, C. A. Goble, R. D. Stevens, J. Zhao, M. Addis, D. Marvin, L. Moreau, and T. Oinn. "Provenance of e-science experiments—experience from bioinformatics." in *Proc. UK e-Science All Hands Meeting*, 2003, pp. 223-226.
- [15] M. Szomszor, and L. Moreau. "Recording and reasoning over data provenance in web and grid services." in *On the move to meaningful Internet systems 2003: CoopIS, DOA, and ODBASE*, 2003, pp. 603-620.
- [16] K. Hamadache and P. Zerva. "Provenance of Feedback in Cloud Services."
- [17] K. Hamadache, and S. Rizou. "Holistic SLA Ontology for Cloud Service Evaluation." in *Advanced Cloud and Big Data (CBD), 2013 International Conference on*, 2013, pp. 32-39.
- [18] SmartSkiGoggles, Evolaris [Online]. Available: <http://www.evolaris.net>
- [19] L. Moreau, P. Missier, J. Cheney, and S. Soiland-Reyes, "Prov-n: The provenance notation," W3C, W3C Candidate Recommendation 11 December 2012, 2012.
- [20] J. G. Breslin, S. Decker, A. Harth, and U. Bojars. "SIOC: an approach to connect web-based communities." *International Journal of Web Based Communities* 2, no. 2, pp. 133-142, 2006.
- [21] "The friend of a friend (foaf) project," [Online]. Available: <http://www.foaf-project.org/>
- [22] F. Bendix, R. Kosara, and H. Hauser. "Parallel sets: Visual analysis of categorical data." in *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pp. 133-140, 2005.
- [23] R. Hoekstra and P. T. Groth. "PROV-O-Viz-Understanding the Role of Activities in Provenance." in *Provenance and Annotation of Data and Processes: 5th International Provenance and Annotation Workshop*, 2014.