SmartReality: Augmented Reality + Services + Semantics

Submitting authors:
Lyndon Nixon, STI Research GmbH lyndon.nixon@sti2.org
Jens Grubert, Graz University of Technology grubert@icg.tugraz.at
Gerhard Reitmayr, Graz University of Technology reitmayr@icg.tugraz.at

Abstract: The SmartReality project is a new Austrian national project dedicated to exploring how people can access smarter information about the things around them through a combination of Augmented Reality, Web services and semantic technologies. This paper introduces the project and our initial thoughts concerning the current and necessary state of AR standards, interfaces and platforms.

1. Introduction: SmartReality

SmartReality began life in October 2010 as a nationally funded project in Austria. The participating partners are STI International, Technical University of Graz, Seekda GmbH and play.fm GmbH. Together, they represent expertise and innovation in AR, semantic technology, Web services and online media platforms. In the context of the project, they will explore how people may be able to access relevant information, content and media about objects of interest in their vicinity via Augmented Reality.

A general illustration of how SmartReality would work is provided below:

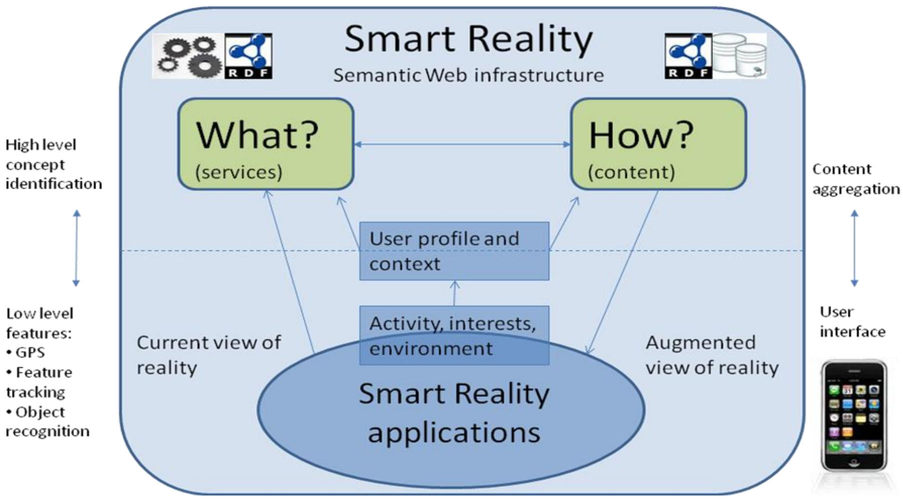

Figure 1. A general view of SmartReality

1. The current view of reality is captured in the augmented reality application according to the video frames (from the smartphone camera). This is extended together with location information from the phone (GPS) as well as other user-centred information which is provided in advance (current activity, personal interests, actual environment) and passed to

server-side services to analyse the video frames and provide object recognition (based on prior stored images in a server-side database). Phone-side the application is able to track features of the recognized objects to ensure a fluid overlay of augmented content over the object (i.e. as the user moves the phone, the augmented content remains displayed in proper relationship to the related object feature).

2. The “What?” of SmartReality will be addressed by a services framework which makes available sets of Web services for the provision of appropriate content and functionality to user’s according to their current context. Service selection and composition is based on one hand on the recognized objects in the smartphone camera view, and on the other on the use of annotations of the services which provide a machine-processable description of their functionality, and relevant object types. This step is further refined by information relating to the user’s interests, context and environment.

3. The “How?” of SmartReality will be addressed by a semantic data infrastructure on the Web (both existing and created by the project), in which relevant domains are modeled formally using ontologies and then relevant content is annotated according to these ontologies. Developed services will query semantic data stores and retrieve relevant Web content, which is packaged and delivered to the smartphone application in an appropriate form.

It is already clear from this diagram that different types of data need to be exchanged or accessible via APIs, mainly Web based, to allow different components, both inherent to the SmartReality platform itself as well as re-usable external interfaces to different data sources and Web services, to realize together a SmartReality application.

2. A SmartReality architecture

As a first step in the project, we have been considering what this SmartReality vision means in terms of the software architecture, and in particular, how it differs from the “classical” architectures of today. Considering the current set of AR application offerings, like Wikitude or Layar, one can visualise the architecture as below, with a browser in the client device providing the usual AR functionality (UI with camera access, location and orientation tracking via GPS and compass, content rendering etc.) while a server-side platform allows developers to create and define their AR worlds (or layers) and hosts the definitions on their behalf, handling the client device accesses.

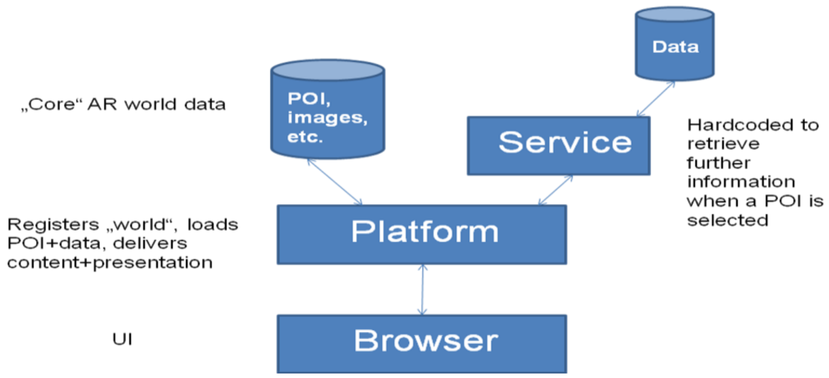

Figure 2. A classical Augmented Reality platform architecture (Wikitude, Layar etc.)

In SmartReality, we find that it is important to be able to flexibly deal with data around a wide range of different object classes, and also build dynamic processes around retrieving and processing that data in the correct manner to support a wide range of different types of SmartReality scenarios. Hence we foresee that a single SmartReality platform[1] becomes necessary to support the client device. For example, the platform handles semantics-based reasoning over an aggregation of data from different sources and provision of content filtered, ranked and prepared for presentation to the client.

The platform will need to encapsulate logic about what sources to take at each stages of realizing SmartReality .That is, the objects of interest, related metadata, services, content and media - as well as templates/rules for how to handle the aggregated data at each step which are relevant in the context of the particular application. These steps may be further defined and refined through the merging of available models about the user (their interests and preferences), context (e.g., activity, location, time) and presentation constraints (e.g., device and network characteristics, user capabilities).

All relevant data for SmartReality can be made available to the SmartReality platform via Web services. By using a services repository the relevant Web services can be registered with their machine-processable descriptions allowing a dynamic look-up of relevant data and services by the platform at any time. This is in contrast to hardcoding to specific interfaces which may change, go offline etc.. It should be possible for the platform to ask for certain types of objects close to the user position in this abstract manner (“*all objects of type T in a 1km vicinity of location (X,Y)”*) .

The service calls will provide platform access to functionality for each step of SmartReality, as detailed in the newly generated architecture diagram below:

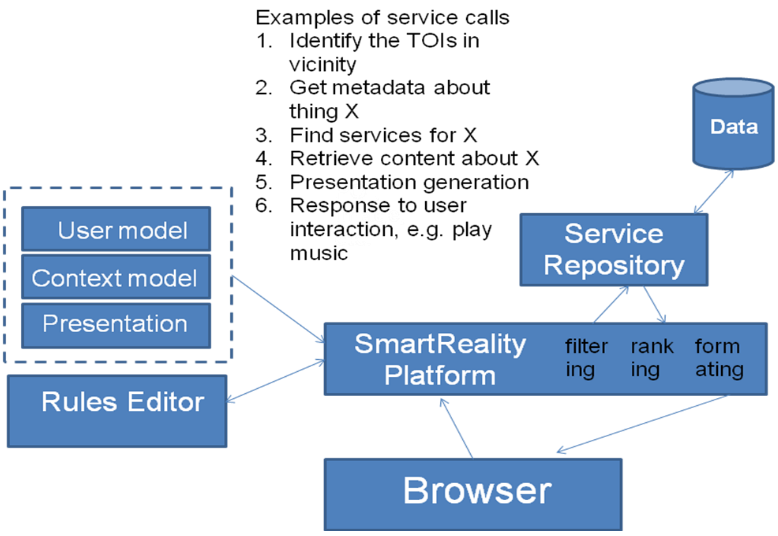

Figure 3. Our vision for a SmartReality architecture, with functional steps realised via Web services

[1] Needless to say, a single platform in the abstract view could be in the concrete implementation anything from a single server-side application to a set of Web services distributed across servers to a virtual platform running in somebody else’s „cloud“.

3. What SmartReality means for standards

It is clear from the above SmartReality architecture that various aspects of SmartReality need to be defined and specified in – where possible standardised or standardisable - formats: data types, various models, service interfaces, rules. SmartReality will explore possibilities in terms of existing standards or specifications, and where the current approaches prove not to be sufficient, will be active in proposing extensions and new specifications. Important prerequisites from our point of view are:

(i) Backwards compatibility with existing AR specifications, so that existing data created for Augmented Reality could be easily repurposed for SmartReality;

(ii) Re-use of existing Web services, data and metadata, hence a support for the specifications and standards of the current (and emerging) Web driven by the W3C;

As such, we elucidate our first impressions and intentions in prototypically implementing the SmartReality platform in this year and developing a set of exemplary scenarios following the SmartReality architecture. A summary of the resulting proposals for specifications and standards for SmartReality is illustrated below, we then turn to each area in turn: Points of Interest (POI), POI metadata, Web services, AR browser UI, and users & context.

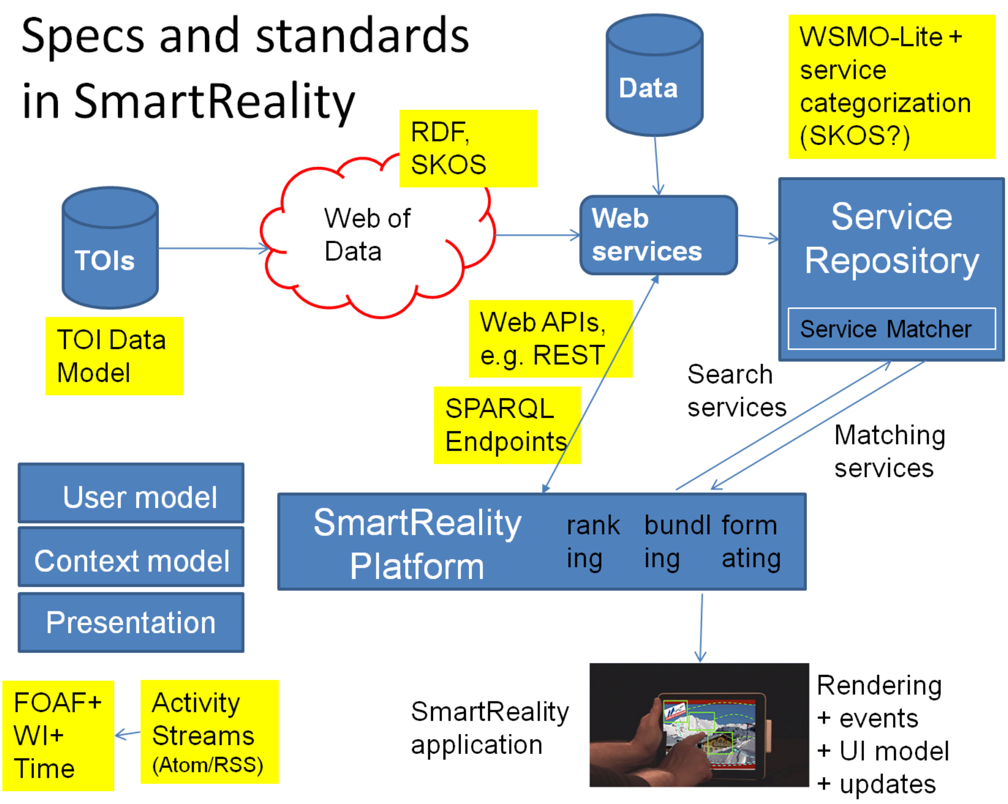

Figure 4. Specifications and standards mapped to Smart Reality

3.1. *From Points of Interest to Things of Interest*

In contrast to a typical focus on locatable "points of interest" in AR applications, SmartReality considers any object in the user's vicinity to be potentially relevant. Current POI approaches tend to combine all the data about the POI in the singularly retrieved POI data set, i.e. not just the core information (POI's GPS location for example) but also the (basic) metadata and associated content for that POI. In SmartReality, it suffices to be able to identify what the thing *is* in such a manner that further metadata or associated content about it can be retrieved from other sources.

Hence, we propose an extension of the POI data model to **TOI (Thing of Interest)**, in which it suffices to give the thing a unique identifier and a statement of belonging to a particular class of things (where the class identifier may also be considered as unique). To allow for the detection of the thing within the AR browser, location data may be provided or visual feature descriptors (or any other means for detection). Such data could also be "outside" of the core TOI description, and instead dynamically associated to it at retrieval via the look-up of further metadata based on its unique identification (so that, e.g. the current location of moving things can always be found).

In terms of the core POI model currently published by the W3C POI WG [1] a Thing of Interest would need fundamentally only a name, identifier and category, where the latter two are identified by unique URIs (see section 3.2. below).

3.2. *Data about things from the Web of Data*

As mentioned above, things would need to be uniquely identified in such a manner that further metadata and data may be retrieved about them. While local databases could be set up for such purposes, SmartReality intends to explore the possibilities to re-using structured data already published on the Web (or which could be published by anyone on the Web following agreed procedures, in order to allow it to be re-used in various applications, including but not only SmartReality). Directly linking to full HTML pages about things as is often done now in AR applications is clearly unsatisfactory as it forces the user to switch to a different application, e.g. a full screen Web browser display. In any case, since the link is direct to the Web content, there is no flexibility on the application side to select relevant content and display it.

In order to enable in SmartReality a more flexible handling of information presentation in the AR browser, we propose to re-use **Linked Data** [2] as a specification for publishing and retrieving data about things via the Web. Basically, any thing can be given a unique URI and metadata about the object retrieved by a HTTP request with RDF/XML response on the object's URI [3]. As an example, Wikipedia has been published as Linked Data under the name DBPedia and metadata about many things which have a Wikipedia page. It can be retrieved via the respective DBPedia URI:

http://en.wikipedia.org/wiki/Eiffel_Tower (the Wikipedia page about Eiffel Tower)

->

http://dbpedia.org/resource/Eiffel_Tower (the DBPedia *resource* about Eiffel Tower)

Note that Linked Data resources also link to further resources such that applications could browse resources along certain properties and there are Linked Data services also to return other equivalent resources (i.e. other descriptions of the same Eiffel Tower from other sources).

By using Linked Data, SmartReality can re-use already existing standards and technologies from the semantics community such as RDF (data model) and SPARQL (query language). Metadata about things of interest can be crawled from the Web, or retrieved from Linked Data caching servers which already provide interfaces to the main Linked Data sources (like DBPedia). Specific sources can be chosen according to knowledge about the type of thing being discovered, using the TOI's class identifier.

Classes can also be URIs pointing into Web-based ontologies (models of domains about things) from which also more general or specific classes can be discovered. For example, in the DBPedia it can be seen that the Eiffel Tower belongs to the class of things which are Buildings, Places, TowersInFrance, SkyscrapersInParis, and even PlacesWithRestrictionsOnPhotography.

It is a subject of research how the existing scale of structured data being published online on the Web via Linked Data can be utilised most appropriately in automated systems to discover information about things, which SmartReality will also pick up on and explore.

3.3. Services to link virtual objects back to the real world

Once we know something about the object of relevance, we feel that in SmartReality it is important to be able to link the virtual object represented in the AR browser back into the real world. After all, the object seen in the AR world is representative of something in the real world and capable of supporting some real world interaction. A typical example would be a business represented in the AR browser is also a commercial entity able to sell products or services to the user. Hence, in SmartReality we aim to describe services offered by things and link these to Web services, i.e. means to perform those service interactions via the Internet. "Point-of-interaction" service offers are of great interest to many domains, from businesses which would like to sell things via the AR browser as soon as the user has selected them in the AR browser view, to commercial opportunities linked from the thing's type such as linking music or a painting to purchasing a song or an art book.

Hence, Web services should be described using machine processable descriptions, where the same categorization schemes and identity model can be re-used from Linked Data (hence the thing's type can also be used to search for services relevant for that type, or generalisations of that type). Additionally, the service's type in terms of a categorization of *what it does* will be researched in SmartReality, since services around a particular domain (e.g. an event) have different relevance in different contexts, e.g. if the event is past, an event booking service is no longer necessary.

For service descriptions, we will explore the use of **WSMO-Lite** [4] and for the description of product and service offers related to things, we will investigate the **GoodRelations** [5] ontology.

3.4. *Cross-device asynchronous models for content presentation, media embedding, interactive objects and in-browser events*

The AR browser needs to be able to render the content generated by the SmartReality platform. We expect to prepare content on the server to reduce the load on the device, so the main requirement for the browser is to be able to receive and render content of different types. The browser will handle the core functionalities for AR, such as access to the device camera, (partly) object detection, pose tracking, and user interaction.

We have noted that since we want to work with different data formats in standard Web and document serialisations such as JSON and XML, we prefer to be able to pass such serialisations to the end device such that a basic adaptation or selection between different elements could still be performed at the time of rendering on the device. However, just rendering typical 2D content formats including HTML(5), SVG (synthetic objects) or SMIL(multimedia presentations) in a 3D augmented view is not always the ideal solution. Perspective rendering reduces the available pixels and adds distortion, while the continuous user interaction of moving the camera disturbs reading text and image details.

Therefore, an intermediate transformation layer that creates appropriate visualization items could be useful. Based on the available content formats for a specific resource and the device specifications (such as screen size, interaction mode) the system could select and transform incoming media. Furthermore, it could generate level-of-detail representations that switch from simple icons, signifying the presence of information, to more detailed views including the 2D media formats. Such an abstraction of the data can, where necessary, be fully handled within the platform with a formatting engine able to generate on the fly end device content instructions for passing to the browser (e.g. using 3D scene descriptions rendered using OpenGL ES libraries).

Even then, a lot of in-browser interaction requires only partial updates of the content displayed in the browser and this greatly facilitates rich browsing on the small device screen. Hence it is desirable to have a end device presentation model able to accept only “updates” of the on-screen rendering rather than necessitate full reloading of the content to present at every change. This can be compared with the shift in capabilities of Web pages made possible through AJAX (Asychronous Javascript and XML), while a part of a Web page may be changed on the fly without needing to retrieve the full Web page content from the server. To allow for this, it seems necessary to be able to have access to an AR browser Document Object Model (DOM) as well as allow client-side scripting, as in the AJAX model. We foresee wanting to make calls to Web services directly from the browser (e.g. using the REST model) and define content updates from within the application itself (extract content from the service response and display in a particular browser “display area”).

Finally, we note that our browser application will benefit from standardisation of the AR browser interaction and event models, e.g. “object selection” by user action on the device, or a “in object vicinity” event thrown by the application itself.

3.5. Users and context in reality-based applications

As shown in the SmartReality architecture, both modelling the user and context may prove important to allow a SmartReality platform to better filter and rank data aggregated from different sources for delivery to the AR browser. We mention briefly here possible specifications which we will use as a starting point for our exploration of these points:

(i) FOAF+WI as an user model

FOAF [6] is a simple property-value model for describing people, generally intended as a means of interlinking people on the Web of Data. As such, it can be used to model social networks, workplaces etc. While obviously not as widely used as the models in social sites like Facebook, many in the semantic community have created FOAF profiles and others have explored modelling public user data in FOAF, e.g. wrapping LinkedIn profiles. The FOAF model has also been extended with Weighted Interests [7], allowing a user's interests to be modelled in terms of Linked Data subjects and weighted on a standard scale. Hence relevant content for the user may be determined and ranked on the basis of this semantic information, even when the explicitly stated interests of the users and the subjects of the content in question do not directly match [8].

(ii) Activity Streams for capturing context

While user's can help define interest models explicitly to allow AR browsers to give more personalised selections of Things of Interest, and the associated content and services, there is also the aspect of implicit determination of relevance of things at any time in the AR application. We feel in AR this is particularly relevant because the user is moving around and their situation is changing in what is essentially a real time browsing experience. This also requires modelling if the selected contextual information is to be available to the SmartReality platform and used in filtering, ranking or presentation steps. There are different aspects of context which could be mentioned, including location (generally provided from the device via its GPS information) and time. Reasoning models may also need to be defined for interpreting contextual information (what does a particular location or time actually mean in terms of the current data under examination?). We mention here another interesting use case for capture of context, based on the ability to acquire and reason over user activity data. While it may be too far-fetched now to consider use of additional device sensors to e.g. determine the user's mood or what they are doing, and avoiding here the understandable privacy debate, we could proposition that two sources may be available for more richly extracting user context:

(1) Determining context from what other actions the user performs on the device

Many users already opt-in to letting applications know where they are. How about opting-in to let applications know they are listening to music or writing an email or searching for 'concert tickets' in their search engine? While this may encounter privacy-issue barriers, if an AR browser can know what an user is doing with their device at any time, it could then provide them with better contextualised AR browser content when they switch back to the AR application. User device activity would need an appropriate model which can be "streamed" to the SmartReality platform in a privacy-preserving manner, identifying the application, possibly its type, and some generic indication of the activity type performed in it.

(2) Determining context from the user's public activity on the social Web

Another option, where users permit access to social networks such as Facebook or Twitter, is using recent Social Web activity by the user to derive a context. E.g., potentially some users tweet about what they are doing via their mobile device, and those same tweets could be fed back into the AR application to contextualise the browser view. Twitter already provides time and location information with tweets allowing to spatially and temporally track a user via their tweets, and the Twitter Annotation API [9] will allow for other, potentially relevant, information to be marked up in a tweet and re-used by tweet-processing applications. For example, by linking a named concept in a recent tweet by the user to the presumption the user is currently interested in things about that concept could lead to the AR browser providing a view of the user's reality centred on that concept.

In regard to both points, we mention the Activity Streams [10] activity which provides a model for syndicating user's activities in social sites around the Web (and could be extended to capture user's activities in mobile apps, for example) and whose resulting data streams could be used in real time by a component in the SmartReality platform to derive additional contextualisation steps in the SmartReality application.

4. Conclusion: a SmartReality future

As described in this paper, SmartReality is about advancing augmented reality applications, moving away from the hardcoding of POIs and their associated (meta)data, and enabling more dynamic and flexible AR experiences which can gather data from different sources, aggregate and process it using semantic models to richly discover links between concepts, classify things of interest, and use available sources to filter and rank an overload of information, add personalisation and contextualisation steps, and deliver final content steps in dynamic and richly interactive ways to the user of the SmartReality application.

In a first examination of what this means for AR standards, SmartReality seeks to (re-)use specifications and standards exploiting widely used, Web conformant and flexibly interchangeable data formats and models, hence highlighting XML as a data interchange format, RDF as a lightweight data model for semantic information and JSON as a lightweight Web application data model . We also seek to conform to current W3C efforts, such as the POI WG in Augmented Reality but also from other domains such as semantics (Linked Data), services (WSMO Lite), mobile (Mobile Web Initiative) and media (Media Annotations WG). In particular, we seek to extend current considerations about POIs to a more generic TOI – Thing of Interest – linked into the existing and growing Web of Data. Also services and content sources on the Web need to be linked into this new infrastructure, in order to enable end-to-end functionality for content retrieval and delivery to the end application. Many interesting initiatives are also just beginning, from which SmartReality hopes to take inspiration and to which SmartReality plans to feedback extensions, alterations and suggestions from real application usage experience. In this light, SmartReality hopes to both learn from the wider discussion on AR standards, and act as an adopter of new standards and tester of proposals, as well as generate its own proposals and specifications in collaboration with the AR community in the next two years of implementing and prototyping SmartReality.

References

[1] http://www.w3.org/2010/POI/wiki/Core/Draft as at time of writing, 17 January 2010

[2] http://linkeddata.org/

[3] http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/

[4] http://www.w3.org/Submission/2010/SUBM-WSMO-Lite-20100823/

[5] http://www.ebusiness-unibw.org/wiki/GoodRelations

[6] http://xmlns.com/foaf/spec/

[7] http://smiy.sourceforge.net/wi/spec/weightedinterests.html

[8] http://www.slideshare.net/laroyo/cnr2010-loraaroyo

[9] http://dev.twitter.com/pages/annotations_overview

[10] http://activitystrea.ms/