Narrating Annotated Experiences

A half-way summary

10-11-2009

Bastiaan van der Weij
Philo van Kemenade
Jan Overgoor

Introduction

We are working on a mobile system that generates a story out of a large set of available location-bound experience expressions, left behind in an Augmented Reality environment by a collection of users.

Augmented reality

The term 'Augmented Reality' (AR) is used to describe the method of viewing the (real) physical world around us, together with an added (augmented) layer of digital imaging on top. This layer of virtual information is closely linked to the actual reality you perceive. In this way it augments the reality that was already viewable, with an extra layer of semantically connected and useful information for the user.
Any device with a camera, a screen and enough computational power, can, in theory, be used to enable an AR view. At this very moment, as more and more people use smart mobile devices, a large amount of people are enabled to use this way of viewing the world in their everyday life. In our project, our aim is to develop a location-based story environment that makes use of experience expressions as material for augmentation, designed for the Android mobile OS.

Grafitti application

In developing our application, we base our work on the 'Grafitti' project previously done by Artificial Intelligence students at the University of Amsterdam [Ref]. This project resulted in a working application that enables users to explore reality and enrich (augment) it by leaving virtual grafitti marks. These virtual AR expressions are stored in a database and can be viewed by other users that come across the location where they are left.
Looking ahead, we expect that soon even more people will own a smart mobiledevice with AR capabilities. If applications like the 'Grafitti' app become popular, people will leave their personal expression all around and you could expect a popular, central square, such as the Dam in Amsterdam, simply to be flooded with AR expressions. Therefore it is interesting and relevant to think of a way to filter all available expressions so that the interesting ones can be presented to the user (personalisation of data).

Layar

An application that makes use of AR as its essential functionality, is Layar. [Ref] This 'Augmented Reality Browser' offers multiple layers to choose from in order to augment reality with useful information. Results for items that are listed in a layer are presented on a grid overlaid on the camera's reality view. Examples include: nearest ATM's, GPS-tagged pictures and historical information.
An AR browser like Layar is an ideal environment to store, create and explore virtual grafitti expressions in. That is why we will make use of this environment.

Goal

Our goal in this project will be to develop an application that presents to the user a sequence of available AR experience expressions, in such a way that this follows a story structure.

User

An application described above will be most useful for people that like to engage in city exploration, such as a tourist or any person that likes to discover nice spots across a city. We describe our user by the following characteristics:

- He / she explores the city
- He / she wants to see places
    - that are popular
    - that are interesting now

Locations

Considering these characteristics, we thought of the locations along which our story sequence should lead. We coin the term 'hype spot' as a location around which are many AR expressions and preferably recently created (within the last days, depending on the amount). We choose to lead our story sequences along such points for several reasons:

- They give a rich base to choose expressions from
- Link to reality
    - many people leave an expression at this location, so it must be popular
    - expressions are 'fresh', so something happened here recently

Story Engine

The story engine takes a set of available expressions and returns an ordered subset of them. The selection and ranking complies to a few restrictions:

- The demand the story lays on the user. This includes the amount of time and walking distance it requires to complete it and the complexity of the story (by the number of expressions used and the maximum number of nesting sub-arcs).
- The sequence has to feel like a story. This means that there has to be semantic coherence and continuity. We use a generic story type model and connect the

expressions using their annotations. The story also has to follow certain narrative theory laws, for example, by using tension bows (see story structure below).

User demand

The demand on the user will be regulated by simple equations that calculate the minimal and maximal number of expressions used in the story depending on the set time-limit or destination.

Story structure

We chose to represent a story as a collection of (possibly partially overlapping) sub-arcs that can recursively be regarded as stories in their own right. A story element consists of a starting and a closing element and an arbitrary number of elements in between, that can be either atomic expressions or sub-arcs. This representation has evident computational advantages. First, it divides the task of finding a good sub-set in a possibly very large set of available experiences into smaller sub-tasks, namely to find a good sub-arc. Second, it is more efficient to reason over composite elements than over larger sequences of atomic expressions. Third, it becomes possible to define generic story types as containing a certain number or sequence of sub-arcs. For this to work, however, two blank spots have to be filled in:

- How can it be ensured that a sub-arc internally makes sense?
- How can the sub-arcs be combined so that global continuity is maintained and the story stays engaging?

The first step to ensure both requirements is to formulate a definition of the story type and the possible sub-arcs. Both of these require a set of well-designed and available meta data, which is the second step.

Story Type

In this project we chose the Odyssey as a generic story type. It has the following advantages:

- It's classic
- It's well known
- It's an epic voyage; this maps well to our sequence along a route
- The story sub-arc structure is generic

It is therefore the perfect story type to generate stories of arbitrary length and elements in it, while still keeping a global continuity.

Sub Arcs

One task is to define the possible sub-arcs. We haven't done this formally yet, but we've been using a list of examples for the time being, consisting of:

(for the Odyssey story type):
- Setting in motion / coming back

- Climax

(for the generic sub arcs):

- Obstacle / block / distraction / set-back
  - Monster / battle
  - Quest
  - Love / friendship
  - Illusion / deception / seduction

Meta data

We distinguish three types of annotations that the creators of the expressions and other users that might perceive them can create:

- *Denotative* - represents what the expression denotes. This can be used to create semantic coherence, or fit an expression in a certain sub-arc on basis of its contents.
- *Connotative* - represents the subjective experience that gave rise to an expression. This can be used to create tension bows, or fit an expression in a certain sub-arc type, for example 'battle'.
- *Narranotative* - represents the use of the expression for a narratative purpose. This can be used to build up a story that makes sense.

The last type, the narranotative annotations, is the most important of the three. The more specific they are, the more able the system will be to produce well-crafted stories. We think that the best format to acquire narratative information about the available expressions is to provide the user with a predefined set of tags. Too much of this effort will, however, lose the purpose of the system. The goal is to find a balance between the amount of required user input and the amount of generating that the system has to do.

Progress

We've laid the conceptual groundwork and have been working into formalizing the ideas, definitions and formulas for the story engine. One week ago we've set our first steps for the implementation. We've been exploring the Android framework, as we're building an extension to the existing Graffitti application. At the moment we're implementing the required meta data structures and a way of viewing multiple expressions in sequence, in extension to the single expressions viewer of the current application. Parallel to that we're working on the story engine, which, after finalizing the definitions, should not take long to implement. In any case, it will end up being a work in progress as one could include as many ideas as they want in the design of the story engine. We're aiming to include at least our basic ideas about the structure of the generated story and leave a discussion of the possible extensions for the final report. Possible future extensions can include for instance an integration of a connotative structure to the story and a more interactive generation of stories that can handle real-time choices of the user.