UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

CAPTURING ROUTE EXPERIENCES

Nikolay Stanchenko

DISSERTAÇÃO

MESTRADO EM INFORMÁTICA

2014

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

CAPTURING ROUTE EXPERIENCES

Nikolay Stanchenko

DISSERTAÇÃO

MESTRADO EM INFORMÁTICA

Thesis advised by Prof. Doctor Tiago João Vieira Guerreiro
and co-advised by Prof. Doctor Luís Manuel Pinto da Rocha Afonso Carriço

2014

Abstract

Several systems currently exist to support creation of location-based stories and capturing of life experiences. However, it has been shown that there is a trade-off between fully committing to the authoring process and "staying in the moment", which produces strain and increases authoring effort.

The present work addresses this problem by leveraging the large amount of third party content, readily available through various web services. More concretely, we re-imagine in-situ storytelling process, providing authors with *suggestions* of external story elements, such as Foursquare[1] venues, which they can embed directly into the story. We explore whether with this approach authors are able to balance between producing novel and reusing existing content, saving time and effort whenever necessary.

Results from our two user studies suggest that *suggestions* can potentially reduce the authoring effort, but only provided they are relevant enough. At the same time, they can significantly improve the viewing experience, provided they are content-rich: Foursquare venues, encompassing photos, reviews and comments, are a good example. We also found that authors valued stories' individuality more than viewers, as the former were somewhat reluctant to "dilute" their personal content with external data, whereas the latter appreciated the social aspects contributed by *suggestions*.

Keywords: storytelling, route, media, web services

[1] `http://foursquare.com`

Acknowledgements

I am indebted to: *Tiago Guerreiro* for his mentorship, guidance and enthusiasm; *Luís Carriço* for his invariably good spirit and amazing Irish dancing skills; *Diogo Marques* for his critique and consistent resisting of the route storytelling concept; *Kyle Montague* for putting his heart and soul into testing the repeatedly crashing prototype; *Henrique Figueira* for his support and *RouteStory* icon; all the participants of all our studies; *Scala-on-Android* community for using and contributing to my open-source projects. Finally, all my friends and family for bearing with me.

Contents

Chapter 1

Introduction

1.1 Motivation

Sharing everyday experiences is an important part of human life [1]. With the advent of mobile technology and social networks, this task has become easier as ever before, but are the results as fruitful, as we would like?

In this work we focus on location-based narration of experiences, as places and routes have historically played great role in the stories we tell: from the myths and legends of Australian indigenous people [25], to modern travel blogs and geo-tagged posts on Facebook[1]. Unfortunately, most current systems do not fully utilize the entire route's narrative potential, only keeping references to its single points. Moreover, location-based storytelling poses an important challenge, as the technology gets in the way while we strive to both share and live through the events of our life.

Recent studies [27, 37] have shown that people may be reluctant to document their life experiences in-situ: sometimes because they want to "stay in the moment", sometimes — because they simply do not have time to interact with the storytelling software. Still, in retrospect people often regret not having captured enough, as the memories fade. Mitchell and Chuah [27] suggest that there is a *"tension between gathering or creating story fragments while experiencing a location, and taking time later to craft these materials into a story"*.

Capturing memories has been famously addressed by Vannevar Bush with his vision of *MEMEX* [7] — a *"device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility"*. Almost 70 years later, many systems have been developed to automatically "log" one's life with pictures, audio, location and biometric data, browser

[1] `http://facebook.com`

and e-mail history, and so on. However, the inherent *"capture everything"* stance has been criticized by several researchers [40], as they argue that lifelogging has wrongly set out to generate an incomprehensible amount of information, rather than complement and aid human memory with meaningful cues. Close to the premise of our work, Byrne [10, 9, 8] suggests that extracting narratives from lifelogs can render them in a more useful form. The proposed solution however relies solely on automatic data capture, without exploiting deliberate story narration in any way.

On the other hand, the rise of social media sharing platforms, such as Flickr[2], has led to the high availability of diverse media content: photos and descriptions of historical monuments, ratings and discussions of restaurants and cafés — which can be retrieved in the context of time and one's location. Consequently, Sueda *et al.* [42], among others, have raised the question: *"Can we describe our own personal experience by using collective intelligence?"* Yet, although possessing a big potential for enriching the stories, such content can easily render them less personal, as vivid in-situ experience is substituted with the sentiments of others.

We believe that the authoring effort may be reduced by taking advantage of both above approaches and incorporating both logged and third party content into the stories, however in such a way that the authors themselves control the proportions of these elements in the final blend. Indeed, our approach allows to save authors' time when certain aspects of the experience may be captured automatically, or when existing third party content may be reused. At the same time, the individuality of the story remains in the hands of the author, as personal content can be easily emphasized.

1.2 Goals

In the present work we aim to explore the process of capturing route experiences with stories, particularly focusing on reducing the inherent authoring effort.

We propose to achieve this through providing authors with suggestions — relevant pieces of readily available content, obtained from various web services, which can be added to the story.

We aim to conduct a study, investigating whether our approach improves the experience of both producers and consumers of route stories.

To this end, we also set out to develop a storytelling system with the following properties:

[2]`http://flickr.com`

- *Route-driven narrative.* In contrast to many existing location-based systems, we utilize the entire route as a backdrop for the narrative.
- *Automatic data capture.* To assist the author in the storytelling process, we provide a way to capture some of the story elements automatically, in the tradition of lifelogging.
- *Third party content.* To further reduce the authoring effort, we present *suggestions* of external elements — that can be added to the story — based on time and author's location.

1.3 Contributions

In the course of the present work we present three contributions:

- *RouteStory*, a new platform for route storytelling (section 4.1).
- *Macroid*, a functional UI domain-specific language for Android (section 4.2).
- A framework for evaluation of storytelling systems (section 5).

Chapter 2

Related work

2.1 Digital storytelling

Storytelling, being one of the oldest, most basic and fundamental art forms, has flourished with the advent of the digital technology. Indeed, stories are now enabled to encompass various media (images, sounds, etc) and allow countless ways of sharing (e.g. social networking) and presentation (interactive web-pages, pico-projectors, etc). In this section we explore the related work on personal, location-based and collaborative storytelling and discuss the research problems outstanding at the frontiers of the domain.

2.1.1 Personal storytelling

Personal storytelling, as a way of capturing one's life experiences, has been a focus of work of several researchers. Appan *et al.* [1] set out to design a framework for sharing life experiences. They indicate that such a framework should place minimal burden on the user, as the users are not expected to *"create long structured narratives per everyday, mundane event"*; they also argue that interactivity is key in discovery and consumption of personal stories: *"We realized that we needed to shift away from the paradigms of cinematic style narrative and passive consumption. Note that this effectively shifts the problem of* authoring structure *in the presentation, to structured interaction, thus leading to greater understanding of the media.* [...] *The users should be able to explore simultaneously, both space and time dimensions of a set of events."*

Kelliher and Davenport [22] developed the *"Everyday Mediated Storytelling Model"*, which is motivated by a six-month ethnographic study with a series of people who solicit, listen or respond to personal testimony as part of their daily job (lawyers, journalists, psychiatrists, detectives, negotiators etc.). An interesting property of this model is that story authors are able to create *'narrative paths'*, connecting their own and others' stories, thus providing interpretation, navigation and discovery cues for the audience. The *Confectionary* application, employing the model, has been further developed and evaluated, proving to be engaging and easy to use.

2.1.2 Location-based storytelling

Blythe *et al.* [3] present an interdisciplinary study of an interactive soundscape, entitled *Riot!* Set in a large public square in Bristol, *Riot!* used GPS to deliver audio samples to individual visitors with relation to their location. The samples made up an interactive play, based on a real XIX century riot. The system is analyzed from the perspectives of *human-computer interaction*, *performance theory*, *museum studies*, and *literary and critical theory*. The paper thus constitutes a valuable example of how digital storytelling systems might be evaluated. The method used is as follows. First, a large-scale questionnaire study was performed with 563 visitors. It captured demographic data, degrees enjoyment and immersion, and other data. Then, 30 semi-structured interviews were carried out, where notes were taken with regard to enjoyment/frustration factors, understanding of the system and the narrative, etc. Finally, 4 ethnographic case studies were performed to gain more insights. The authors note overall positive reception of the system and conclude: *"Location-aware mobile technologies offer more creative possibilities, opening up the psychogeography of public space as a medium for art."*

Gustafsson *et al.* [18] investigate the possibility of enriching pervasive games by providing narrated elements and integrating them with geographical information systems. As a case study, they present *Backseat Playground*, a game to be played by children while driving on the back seat of a car. The game characters reference geographical objects, that could be seen from the window, thus the underlying narrative takes advantage of an otherwise boring activity of getting from A to B. The system's contribution to the field of location-based storytelling is its scalability over vast areas; this is achieved by predicting the player's path, extracting visually available information (e.g. landmarks, intersections, woods, etc) and utilizing this data to produce new events in the narrative.

Mitchell and Chuah [27] set out to provide thematic recommendations to travelers to support them in authoring stories about their trips. Initially they believed the users would want to create the stories as they go, but they found that people often would rather "stay in the moment". As one of the study participants put it, she *"did not want to write captions on the spot, preferring to spend her time 'walking around the streets' rather than thinking about her story"*. However, later the same participant reflected: *"I guess it would be useful to craft it emph[the caption] on the spot because when you go back you might forgot what you've seen or how you felt at that point in time."* This led the authors to believe that there was a certain trade-off between gathering the story fragments in-situ and post-factum story creation. Thus, they suggest that future systems treat these two activities individually.

Most recently, Procyk and Neustaedter [37] share somewhat similar insights into location-based storytelling and games. While they assessed that location provided good grounds for creativity, discovery and reflection over stories, they found that people were often not willing to sacrifice their time, monetary and other resources in order to create or experience a story *on location*. Indeed, many commercially available systems, such as *Foursquare*[1], allow content consumption regardless of the user's whereabouts. They also found that having an established audience, e.g. a group of peers, or a big overall number of users, can stimulate the creative process, while the lack thereof has the opposite effect.

2.1.3 Discussion

As far as developing for everyday storytelling is considered, it has been debated, whether studies of narrative structures such as [38, 5] should be taken into account. Appan *et al.* [1] argue that this would not be appropriate: *"First, it is very hard to assign labels (such as conflict, introduction, resolution etc.) to pictures/writings from every day events.* [...] *Second, constructing causal relationships like supporting events, opposing events, while easily done in fictional narratives is very hard to do with photographs and text that sample the everyday experience."* Nack *et al.* [28] applied the *'Odyssey'* structure to their generated stories, only to find that most story viewers were not immediately aware of it, until told explicitly. We therefore believe that maintaining (or not) a structure of any kind should be entirely delegated to the storyteller. However, as noticed by Obrador *et al.* [30], some sort of structure can be utilized to make the story more balanced, e.g. by grouping similar images and removing near-duplicates. A need for an approach of this kind is further motivated in [9], as discussed in section 2.2.1.

Route and location-based storytelling has proved to be a fertile area for research. We build upon existing knowledge to benefit from the additional depth and narrative qualities that spatial information adds to the stories. As we discuss briefly in the next section, having locations embedded in the story can also aid a particular way of reminiscing, namely reflection and inferencing of one's route patterns and habits, which may as well enrich the narrative.

Finally, several researchers acknowledge the burden that often accompanies personal storytelling software [1, 30, 27, 37]. Obrador *et al.* [30] conclude that *"users consider the task of creating personal photo stories to be* mentally *laborious and time consuming, and this effort is* as high as *their concern to create a good photo story"*; Mitchell and

[1] `http://foursquare.com`

Chuah [27] argue that the moments of *"story gathering"* and *"story creation"* should be studied separately, and further there is a *"tension between gathering or creating story fragments while experiencing a location, and taking time later to craft these materials into a story"*. From these remarks we identify the need to relieve the authoring burden and offer the storytellers more freedom in balancing between "staying in the moment" and documenting their experiences.

2.2 Lifelogging and third party content

The essential premise of lifelogging has long been to automatically capture one's life by various digital means in order to augment and support human memory. In a recent work, symbolically titled *"Beyond total capture"*, Sellen and Whittaker [40] have argued that although early research had been mostly devoted to the *logging* per se, it would be important to explore its actual benefits: *"surprisingly, many lifelogging systems lack an explicit description of potential value for users, focusing instead on technical challenges"*. O'Hara [32] approaches this and other critical works from a broader perspective, based on technological advancements, the nature of mind and our environment; he writes: *"The* [...] *worries, about the collection of useless information and information overload, can perhaps best be appreciated in the context of* [...] *Moore's Law. As noted, the increase in computing power over the last decades has been colossal, and has led to all sorts of unpredictable consequences, of which the feasibility of memory-supporting technology is just one. In general, statistics and number-crunching have time and again proved more useful than cleverer ways of processing information."* Meanwhile, new diverse ideas have emerged, for example, it has been proposed for cars, buildings and other non-human subjects to maintain their very own lifelogs [26].

Using third party content to generate stories is a relatively recent trend, perhaps inspired by the advent of various media sharing platforms, such as Flickr and Foursquare. This approach has an advantage in that such content can be utilized both in-situ during storytelling, or post-factum during story editing, unlike lifelogging, which relies on the information being recorded (even if indirectly so) by the user. The downside is however that the stories employing third party content perhaps might appear to be less personal, than those narrated manually by a single user.

2.2.1 Lifelogs and narratives

Several authors have suggested that lifelog data might benefit from a narrative representation. Gemmell *et al.* [16] particularly emphasize the importance of location and time in empowering both recollection and storytelling. Lindley *et al.* [24] note the change of the way narratives are constructed, which comes with lifelogging. As they point out, since the data is automatically collected, the actual authoring is likely to happen in retrospect: *"The device* [SenseCam] *is unique insofar as images are recorded automatically and so questions about its use become questions about selection* [...]. *One of the features of SenseCam that was valued was its uncontrolled operation, allowing favourable contrast with other recording media such as cameras.* [...] *The responsibilities of 'authorship', then, become almost entirely post-hoc, and the narratives that result, as we have suggested, are occasioned and produced in quite specific ways."*

Building upon the model described by Brooks [5], Byrne and Jones [10] outline *structural*, *representational* and *presentational* considerations, pertaining to creation of narratives from lifelogs. With regard to the first category, they follow [1] to suggest an event-based structure of the stories. The discussion concludes with an open question of whether any formal narrative structures can be applied to such stories. As for the representational aspect, the paper states the need to acquire better context and reasoning from the lifelog events to improve the quality of the narration: *"From our experiences with these retellings, we have found that they are only inherently meaningful to the owner and find little resonance (other than as a result of the relative novelty of the technology) with others. The retelling becoming a passive experience for them. Through this knowledge representation we can seek to include a better understanding of the events and consequently better communicate the meaning, significance and experience of the life story."* Finally, the presentational aspect brings up many questions for further research: How should multimodal digital stories be presented? Should the ever-evolving nature of everyday stories and oral folklore in general be mirrored in the digital counterparts? Should the presentation of the story become interactive, even if that means the story would not be set in stone?

Picking up from [10], Byrne and Jones [9] outline a card-sorting study, where participants were asked to assemble stories from physical representations of lifelog artifacts. While almost 50% of a typical story consisted of *SenseCam* images, the findings suggest that the participants took advantage of different modalities to communicate specific aspects of a story. Yet, interestingly, the strategies and methods they employed were found to be quite consistent. First, most participants reduced the number of artifacts by filtering out the redundant, poor-quality or seemingly irrelevant ones. They then proceeded to select key (focal) frames of the story, forming its central themes. Subsequently, the artifacts were clustered around these themes. The participants however tended to preserve some of the peripheral elements: *"These moments while not core to the plot, were noted by the users to be important to the overall experience. By preserving such elements, the narrative becomes more personal, exemplifying the unique experience of the individual."* Finally, after several iterations of further filtering, the stories were assembled, with the two most prominent arrangements being by time (chronological) and by theme.

2.2.2 Third party content

Nack *et al.* [28] automatically generate image-based stories for tourists to guide them through the so-called *'hypespots'* of the city. As the images are submitted and annotated by the users, the system, named *story-to-go*, clusters them according to the geo-tags and uses a narrative structure to create a route, that passes through some of the clusters. The annotations are used to categorize images and provide themes for the sub-arcs of the narrative.

Sueda *et al.* [42] describe *KiokuHacker*, a system that uses person's GPS logs to produce a timeline, filled with Flickr photos and Twitter[2] posts, found on the Internet. Striving to answer the question *"Can we describe our own personal experience by using collective intelligence"*, they show that such a system encourages better recollection, compared to recollecting "organically".

More recently, Packer *et al.* [34] argue that lifelog data can be enriched by *linking* to the Semantic Web, while narratives can be enriched with data *extracted* from Semantic Web knowledge stores. They present *MemoryBook*, a web interface that produces narratives from lifelogs, performing the said enrichments. The resulting story is event-based, and several ontologies are used to add additional media and descriptions to the events during the phase of *"narrative expansion"*. The following is an example of such expansion (textual descriptions are made out of template sentences; added items are **in bold**):

> *Between 19:05 and 19:20 the 0.3 mile journey to Kai Mayfair was made.* ***The Kai Mayfair is a Chinese restaurant established in 1993, it has a Michelin Star.*** *The 125 minutes between 19:35 and 21:40 were spent in or around Kai Mayfair.*

2.2.3 Discussion

As it stands, a great body of lifelogging research has been dedicated to the capturing, amassing and segmentation of data, as well as to analyzing how this data aids recollection and reflection. Several authors, most notably Byrne [10, 9, 8], explored creation of narratives from lifelogs. This is an important topic, since storytelling largely aids communication of human life experiences (see e.g. [24, 10]). Currently presented systems, however, suggest post-factum, desktop-based workflows, which lack the attraction and utility of the mobile, in-situ solution. Indeed, by coupling lifelogging with narration from the start, we can abolish the user's need to scrape though large archives of automatically logged data at a later time.

[2] `http://twitter.com`

A number of reviewed systems use web ontologies and knowledge databases to enrich the recorded lifelogs or stories [28, 42, 34]. While this approach can certainly contribute to forming more captivating content, it may lead to questioning the authorship and individuality of the story. That being said, Lindley *et al.* [24] point out that even in lifelogging the entire notion of authorship is diminished and blurred, as we enter the digital, automatically captured world: *"digitisation of narrative collapses the distinction between 'producer' and 'consumer', hence democratising authority and expertise.* [...] *A camera that is worn and that takes photos automatically has a number of implications.* [...] *resonating with the weakening authorial voice already mentioned, photos are not necessarily 'owned' by any one individual; there is no photographer."* We suggest that by combining third party content with deliberately narrated elements, a more personal context might be attached to the resulting story.

Chapter 3

Route Storytelling

> *In this chapter we give an informal introduction to the concept of route storytelling, and reflect on the preliminary study we have conducted with an early prototype of our storytelling platform.*

3.1 The concept

We envisioned a platform to improve capturing, authoring and sharing life experiences based on routes. With our system, entitled *RouteStory*, the user travels along the route with a smartphone or a tablet, adding media items, such as photographs, text notes or sounds, at will. In case of images, the camera of the phone or tablet is used; text notes are added simply by typing; for voice notes the microphone is employed. The resulting stories consist of this data arranged on the route trajectory; they are stored locally on the device. Once created, they can be shared and explored by the author or others in several ways: from the traditional map with markers to an immersive 3-D animation.

Our early prototype supported images, text and voice notes, as well as ambient sound, which, honoring the lifelogging tradition, was recorded automatically at a fixed rate.

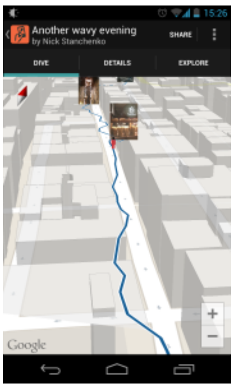

Figure 3.1: An early prototype of *RouteStory*.

3.2 Preliminary study

Aiming to validate the utility of route storytelling and elicit more knowledge about its potential usage, we conducted a two-phase preliminary study, which employed our early application prototype during the second phase.

3.2.1 Research questions

- *RQ1*: What people think about the concept of route storytelling?
- *RQ2*: What features and types of media might be appropriate for a route storytelling system?
- *RQ3*: What are some actual usage patterns and concerns, connected to a prototype route storytelling system?

3.2.2 Methodology

Phase 1

In order to answer *RQ1* and *RQ2* by gaining insights into the possible features and types of media, that could be employed by our system, we conducted an online questionnaire. Not to restrict the participants' suggestions and to allow for a wider audience, we did not expose our research prototype of the system, presenting only a general description of the concept.

We employed a questionnaire that included questions on usage of mobile and map-oriented services, as well as questions asking to rate the benefit of different types of media (e.g. images, audio) for route storytelling. Finally, several open questions were added that called for suggestions regarding features of the system.

54 participants were recruited through e-mail and social networks. The participants were not remunerated.

We have distributed the questionnaire among the participants through Google Drive and collected the results.

Phase 2

In this phase we aimed to elicit ideas based on actual usage of an early prototype of *RouteStory*, thus answering *RQ2* and *RQ3*, and also to assess the user experience of the system and the concept in general.

To this end, we employed two standard questionnaires:

- SUS [4] and
- 10-item short version of AttrakDiff2 [19]

Twelve participants were recruited at the faculty. They did not have any prior experience with *RouteStory*. The participants were not remunerated.

To put the participants in equal conditions, we devised a well-defined story creation task, which remained the same for all of them. In fact, our study environment (university campus) did not provide much room for unconstrained creative route storytelling, and thus our 'artificial' task choice also allowed us to ensure that the system was used to its full potential.

The participants were given an introduction to the system, its goals and main concepts. As motivated above, each participant was given the task to capture the route from the faculty building to the university administration building, with a requirement that the resulting story could serve as a navigational aid for a hypothetical foreign exchange student.

The participants were free to use any types of media available and were asked to *'think aloud'* as the accompanying researcher took notes of any of their concerns. After arriving at the destination, the participants explored their newly created stories and shared their thoughts and opinions on the system and the concept in a semi-structured interview. On average the stories spanned 6 minutes, covering around 300 meters, and demonstrated the usage patterns of different media elements, which we discuss in the next section. Finally, the participants were presented with a questionnaire, comprised of usability metrics, usability assessment of different aspects of the system, and demographic data.

3.2.3 Results

User experience

A high score of 83% was achieved in the SUS [4] usability questionnaire.

Employed media

The participants of both phases considered photographs to be the most relevant type of media for route stories. In particular, stories created in the *Phase 2* each included from 1 to 5 photographs (2.2 on average), making it the most used media type. Interestingly, according to [2], this preference may be specific to Western culture (our participants were Portuguese).

Voice notes, while generally rated as significantly contributing to conveying the route experience in the *Phase 1*, were surprisingly unpopular with those who used the prototype. Several *Phase 2* participants mentioned that they *"felt weird to talk to the phone*", and some remarked that voice notes were *"not helpful in noisy environments"*. At most 2 voice notes were included in a story, with many participants adding none (0.7 on average).

Text notes, despite being rated similarly to voice notes in the *Phase 1*, were much more popular than them with those who used our prototype, and were added to each story up to 3 times (1.6 on average). However, although we initially separated story elements by their type, it was soon evident that most of the users of the prototype missed the ability to attach captions or comments directly to photographs or sounds, in a way mixing several types of media in one story element.

External content

It was noted by several participants of both phases, that the stories would benefit from additional elements. The *Phase 1* provided an extensive list of such elements, including the means of transportation used (*"I think it would be interesting for example for a person who travels around town in a bicycle, bus or train"*), links to web pages, weather data, suggestions for tourists, historical facts (*"It would be interesting if you could give it a historical view* [...] *For example, if a person is around Basilica da Estrela maybe you could show some images of it in the past or maybe some curious historical facts.*"), etc. This went in line with out intention to incorporate external content into the stories.

Privacy and authorship

Although automatic ambient sound recording proved to be useful (and more efficient than manually added elements, with stories featuring 5.4 ambient sound samples on average), several participants of the *Phase 2* were concerned that it could potentially lead to unwanted leaking of their personal life into the stories. While the system did allow to disable this particular feature, we concluded that the situation in general could be improved by providing clear feedback on *what* and *when* is captured automatically.

Looking back at the concept of automatic data capture, we noticed that while it threatens privacy, it (somewhat paradoxically) also diminishes and blurs the notion of authorship, so the stories may appear to be less personal ([24]): *"A camera that is worn and that takes photos automatically has a number of implications.* [...] *photos are not necessarily* *'owned' by any one individual; there is no photographer."* Indeed, with current technology even vehicles and buildings could become 'authors' in their own right ([26]).

Collaborative storytelling

A few remarks collected during the *Phase 1* drew our attention to the possibilities of collaborative authoring as a means of further democratization of route storytelling: *"Another thing that could be interesting is creating a story in conjunction with other people, while they are walking around together. Every participant could provide their own input in their own phone, creating a storyline, segments of which could even be re-arranged later on."*; *"A different user than the one who recorded a specific route, should be able to modify/add to that specific route. For instance, user A records route R. Then user B, who as already traveled this route and has other information or more detailed information about it, should be able to add/correct the recorded data on route R. This could make the story more robust."*

Interestingly, although personal storytelling systems could indeed take advantage of the social, shared nature of many life experiences, very few researchers explored this area, [14] being one of the notable exceptions. While we have not pursued this research direction in the course of the present work, we have strived to take it into consideration when designing our storytelling platform, so that collaborative storytelling could be integrated in the future.

Story structure

Finally, several ideas on providing more flexible story structure emerged during the study. A participant of the *Phase 1* suggested to implement *"route concatenation: it would be interesting to create a 'mashup' of 'stories' by specifying a bigger route that would encompass several smaller RouteStories"*. A similar suggestion for a collaborative setting came from the *Phase 2*: *"For instance, user 1 registers a route from location B to D. Later, user 2 decides to complement that route by adding another from A to B, thus getting a composed story from A to D."*

Additionally, our own experience of using *RouteStory* in the wild — during traveling — identified the need to capture stories in smaller joinable parts, pausing e.g. for meals and overnight stays. Therefore, we decided to adopt a more flexible data model, where stories consisted of *'chapters'*, which could be concatenated (as consequtive segments of the same route) or merged together (as overlapping segments of the same route). However, during the course of the present work we only considered stories with single chapters for simplicity.

3.3 Outlook

As we had bootstrapped our storytelling platform and explored several aspects of its usage, we set out to continue the work towards blending the personal content of the stories with third-party content available through various web services. To this end, we needed to rethink the recording process by presenting suggestions of external media elements that could be incorporated into the stories.

Chapter 4

Design and implementation

> *In this chapter we discuss the design and architecture of* RouteStory*, the storytelling platform we have developed, as well as some technical background on the solutions used, and our open source software contributions.*

4.1 RouteStory

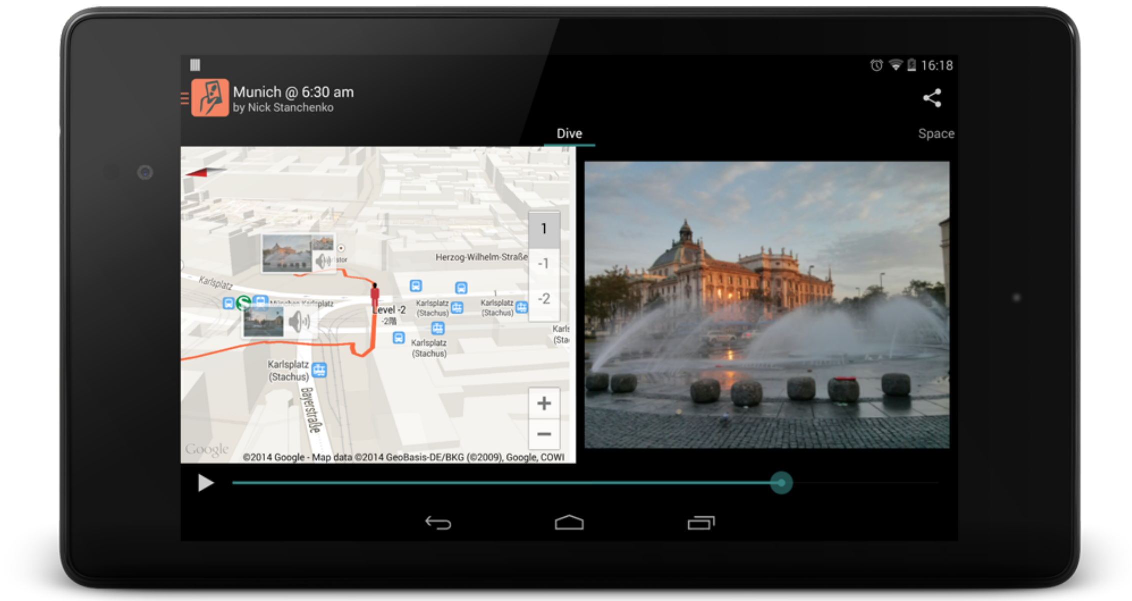

Figure 4.1: *RouteStory*

4.1.1 Overview

Main goals

When designing the platform, we adopted the following high-level goals:

- Develop a system to record, share and view stories based on routes.
- Provide a portable abstract data model for use in the application, server-side or in data analysis.
- Allow importing and exporting stories via regular files.
- Recording-wise:
  - Support adding media elements, such as photos, sounds and notes.
  - Support data logging.
  - Suggest related content from various web services, based on current time and location, which can be added to the story.
  - Allow the system to be extended with additional story element types.
  - Allow the system to be extended to support collaborative storytelling.
- Viewing-wise:
  - Provide a means to explore stories in the time and space domains.

In line with the second goal, we have dissected the system into two modules:

- *routestory-model* contains the abstract data model, serialization utilities, web service clients and several useful algorithms;
- *routestory-android* depends on *routestory-model* and contains the UI code for the Android application.

The main components, on which we will elaborate below, can be thus summarized in this diagram:

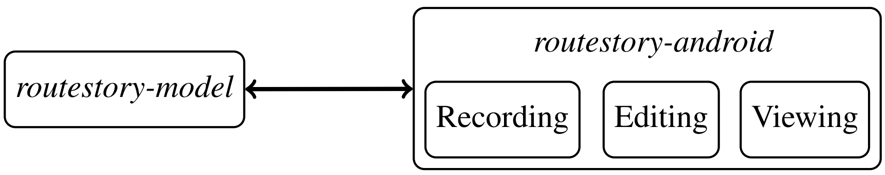

Figure 4.2: Main components of *RouteStory*

Notes on implementation

We chose Android[1] as our target mobile platform, since it is well established and widespread on the market and provides OS-independent developer tools free of charge.

[1] `http://www.android.com/`

We also made a decision to implement our storytelling system in Scala programming language[2], instead of Java, the de facto standard on Android. Scala is a modern language, blending object-oriented and functional programming. It features a powerful type system with type inference, concise and flexible syntax, and a very practical standard library with convenient collections and broad support for various concurrency models (actors, futures/promises, software transactional memory, etc). Ever since, the author has actively participated in the *Scala-on-Android* community, which culminated in developing several open-source libraries and presenting them at the main Scala conference (see section 4.2).

4.1.2 Data model

Schema

Taking into account our early developments and the results of the preliminary studies, we have devised an abstract data model for the route storytelling domain. In this model, stories are represented as collections of chapters. Each chapter contains a list of timestamped GPS locations (the route), as well as a list of timestamped story elements of various types. This representation ensures that all the story elements are arranged along the route — their coordinates are calculated by interpolating the list of locations at a given timestamp. Throughout the present work, we are only using stories consisting of a single chapter for simplicity.

The following story elements are supported:

- text notes;
- voice notes;
- ambient sound (recording is automated);
- photos (with captions);
- Instagram and Flickr photos (external);
- Foursquare venues (external).

The system also supports a special *unknown* element type, which allows us to introduce new story elements without breaking legacy software — they will be simply ignored if not recognized.

The data structures can be more formally summarized with the following BNF notation:

[2] `http://scala-lang.org/`

⟨*story*⟩ ::= ⟨*id*⟩ ⟨*chapter*⟩+

⟨*chapter*⟩ ::= ⟨*metadata*⟩ ⟨*start*⟩ ⟨*duration*⟩ ⟨*locations*⟩ ⟨*elements*⟩

⟨*metadata*⟩ ::= ⟨*author*⟩? ⟨*title*⟩? ⟨*description*⟩? ⟨*tags*⟩?

⟨*author*⟩ ::= ⟨*id*⟩ ⟨*name*⟩

⟨*locations*⟩ ::= (⟨*timestamp*⟩ ⟨*location*⟩)+

⟨*elements*⟩ ::= (⟨*timestamp*⟩ ⟨*element*⟩)*

⟨*element*⟩ ::= ⟨*audio*⟩ | ⟨*image*⟩ | ⟨*text-note*⟩ | ⟨*foursquare-venue*⟩

⟨*audio*⟩ ::= ⟨*voice-note*⟩ | ⟨*ambient-sound*⟩

⟨*image*⟩ ::= ⟨*photo*⟩ | ⟨*instagram-photo*⟩ | ⟨*flickr-photo*⟩

Serialization

The *routestory-model* module supports three (de)serialization formats:

- *JSON*[3] — a universal format, employed by the remaining two. Media files (images, audio) are referenced by URL.
- *zip (.story)* — a simple zip archive, consisting of a single *story.json* file (in the above format) and all the media files. The URLs are thus local to the archive.
- *Couchbase Lite*[4] — a collection of persistence helpers, based on the JSON format and *Couchbase Lite* database. Couchbase Lite is a NoSQL document store with Android support, allowing to persist JSON objects with a simple Java API.

This choice of formats allows us to persist and retrieve stories on the mobile device (using *Couchbase Lite*), send stories between devices via e-mail (using the *zip* format) or expose stories through RESTful APIs (using the *JSON* format).

Web service clients

We have implemented unified clients for three web services: Foursquare, Flickr and Instagram, using our own API client creation library[5], which takes care of caching the required media files and converting the JSON data to our data structures. We used the endpoints which take location as an input parameter to retrieve results, relevant to the user.

[3] `http://json.org/`
[4] `http://www.couchbase.com/mobile#lite`
[5] `http://resolvable.github.io/`

Algorithms

We have implemented a few useful algorithms on our data structures, most notably *route pruning* and *element clustering*.

Route pruning allows to simplify the route curve, producing a more smooth trajectory. This is particularly useful for moving the camera along the route during story viewing, as raw routes usually introduce a lot of shaking. We employed the classic Ramer—Douglas—Peucker algorithm [13].

Element clustering is in fact a naïve implementation of agglomerative hierarchical clustering and produces a tree of elements of a chapter, grouping the more tightly located elements together. The algorithm only uses elements' timestamps to calculate distances between them; although this may lead to neighboring elements not forming a cluster (i.e. if they are at the ends of a U-shaped route), we are more interested in grouping elements which were indeed captured at close moments of time. The resulting hierarchy is annotated with the diameters of the clusters. This information can be used to determine if the cluster should be expanded at a certain scale of the map.

4.1.3 Recording stories

User interface

Recording is undeniably the most important feature of our system. Equipped with a siding view, the user interface provides two perspectives: a map with the current route and story elements, allowing to explore already added content (figure 4.3); and the control panel, providing a means to add new elements to the story (figure 4.4).

The ambient sound recorder is an example of a lifelogging-inspired tool. It is enhanced with the results of our preliminary study in mind: it is fully configurable and provides a clear indication of whether it is switched on; additionally, a vibration is issued every time a sound sample is recorded.

The area below the standard switches is allocated for the suggestions, which are fetched from several web services automatically and can be refreshed at will. Each suggestion can be previewed and added to the story, or dismissed, in which case it does not reappear.

Following either of the above schemes, new element types can be added to the control panel, making our system highly flexible.

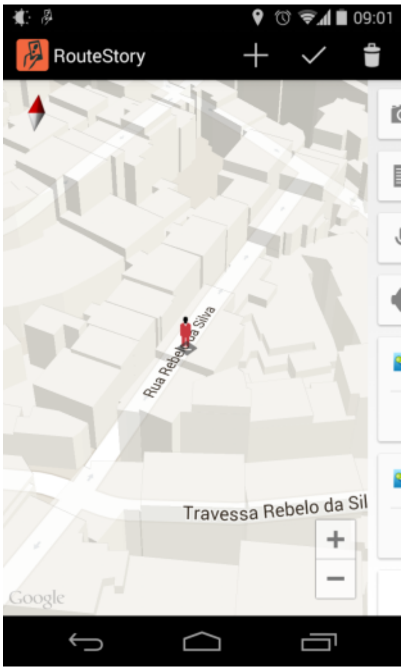

Figure 4.3: The map of the story (empty so far).

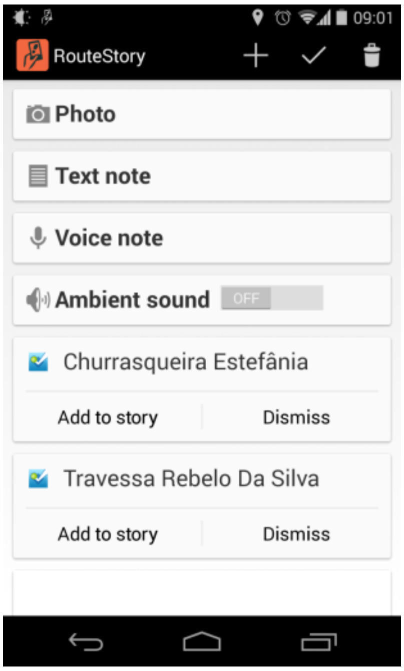

Figure 4.4: The control panel.

Architecture

The cornerstone of our implementation is the Actor model[6][7]. *Actors* are essentially independent components, sharing no common state and communicating between each other with immutable messages. Actor systems possess several properties, which make them a very appealing concurrency model:

- Actors can be mapped to thread pools in any desired way.
- Message-passing is asynchronous (non-blocking).
- An actor is passive, i.e. it only responds to messages.
- An actor processes one message at a time, so mutating its state from the message handler is thread-safe.

Our recording actor system consists of several actors (see figure 4.5). *Typewriter* is responsible for storing the locations and story elements, that it receives — note, that it is safe to submit items to *typewriter* from any thread. *Dictaphone* is an example of actor that produces story content — it records sound automatically in response to a timer, and sends the resulting media to *typewriter*. *Cartographer* is responsible for obtaining the current location and updating the map interface. Finally, *suggester* fetches the suggestions, sends

[6]`http://en.wikipedia.org/wiki/Actor_model`
[7]`http://akka.io`

them for displaying, and maintains a list of the dismissed ones.

Another advantage of employing an actor system is that it can be relatively easily extended to a networking scenario, where it would communicate with another remote actor system(s). Thus, our implementation can be used as a basis for a collaborative storytelling platform.

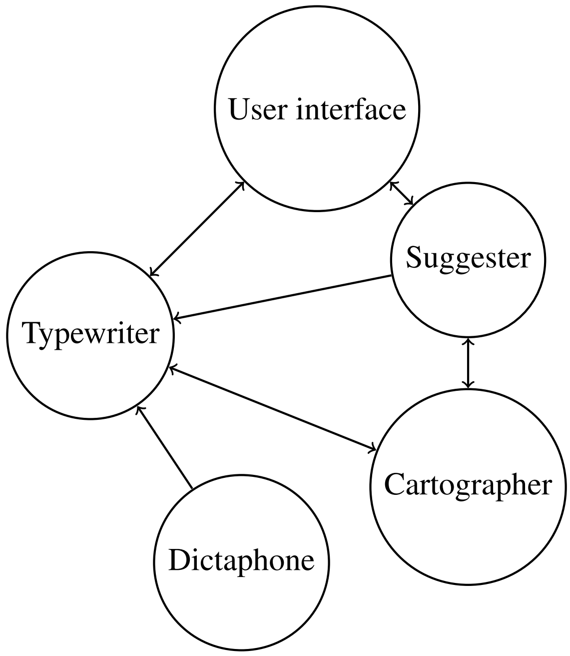

Figure 4.5: An actor system for recording stories.

4.1.4 Editing stories

User interface

The editing interface of our system, while still minimal, provides two important functions: editing the metadata (title, description, etc) and filtering the list of elements — an essential post-processing step for many users [9]. The respective parts of the interface are depicted on the figures 4.6 and 4.7.

Architecture

In the editing mode we employed an actor system, which, although much simpler, is similar to the one used in recording. The *editor* actor is responsible for maintaining the current version of the story being edited. The remaining two actors are used to connect the two interface sections to *editor*.

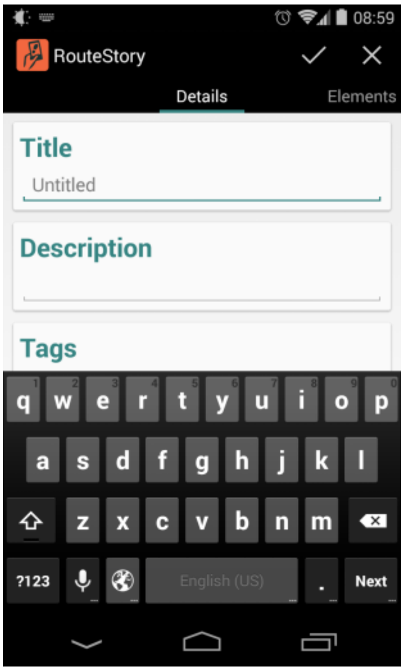

Figure 4.6: Editing metadata.

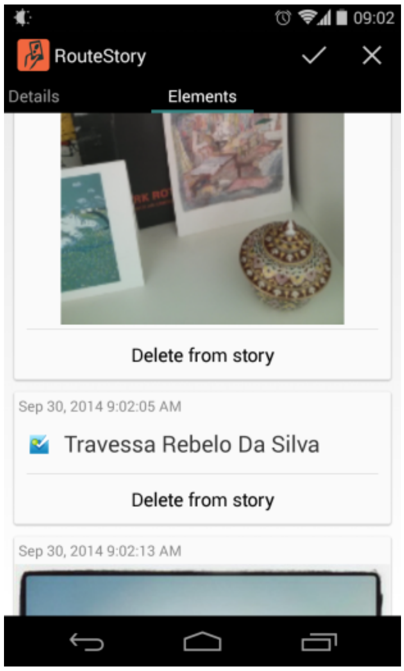

Figure 4.7: Filtering story elements.

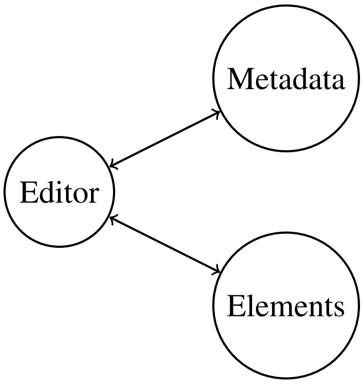

Figure 4.8: An actor system for editing stories.

4.1.5 Viewing stories

An important design consideration for the story viewing interface has been the fact that a single chapter can be visualized in several distinct domains: *time* (element timestamps and progress through the route), *space* (the route and the elements on the map), *time+space* (the route on the map and in time), *element* (navigating between the elements). Additionally, the chapter can be viewed under different magnification levels — this is very important in the *space* domain, since markers on the map need to be grouped together on zooming out and exploded when zooming in.

To cater for each visualization domain, we have developed several viewing modes, entitled *'Dive'*, *'Space'*, *'Timeline'*, *'Element'* and *'Details'*, the latter being responsible for displaying chapter metadata, such as description and tags. It is worth noting that consistent interaction between the viewing modes was as much important to us as the modes themselves. To this end, we had been inspired by the synchronization approaches presented by Gunnar Liestøl in [23].

Dive

In order to allow exploration both in the *time+space* domain, we have developed the *Dive* viewing mode, consisting of a 3D map, overlooking the route with story elements; a preview window, which shows the element in current focus; and a slider, allowing to navigate the story both in time and space. Here we employ *element clustering* and *route pruning*, described in section 4.1.2, to avoid visual clutter on the map and to smoothen the movement of the camera along the route respectively. Note that since element clustering is implemented in a independent and immutable fashion, the resulting hierarchy tree can be reused between several modes, which require maps.

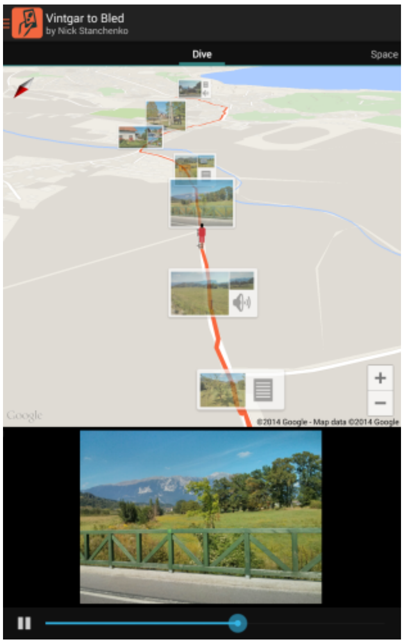

Figure 4.9: The *Dive* mode.

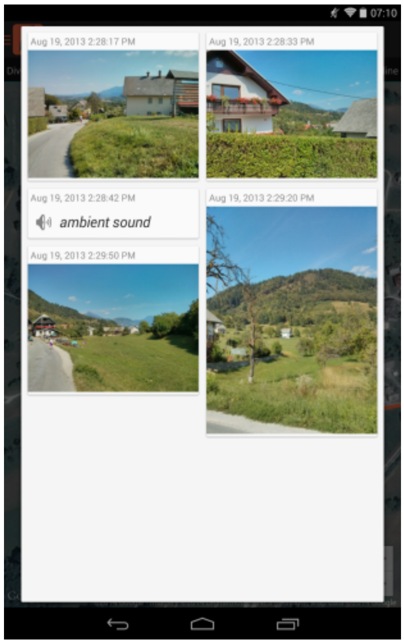

Figure 4.10: The element selection dialog.

Upon clicking a clustered map marker or the preview, the element selection dialog is shown (see figure 4.10), which allows to navigate to a particular element in the *Element* viewing mode (see section 4.1.5).

Space

The *space* domain, conventional for most cartographic applications, was catered for with the respective *Space* viewing mode, see on figure 4.11.

Timeline

To allow exploration in the *time* and *element* domain, we introduced the *Timeline* viewing mode. It features a grid with the previews of story elements, marked with timestamps (see figure 4.12).

Figure 4.11: The *Space* mode.

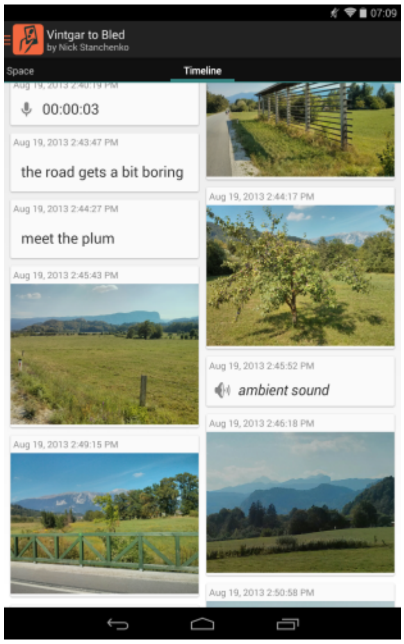

Figure 4.12: The *Timeline* mode.

Element

The *Element* mode (see figure 4.13) allows to swipe between story elements, such as photos, text notes, sounds, etc. Each element is represented as appropriate: in the case of photos, zooming facilities are provided; in the case of sounds, playback controls are added.

Synchronization

A challenging part of the development process has been attaining correct synchronization between the various viewing modes. Here we again employed the Actor model, as it

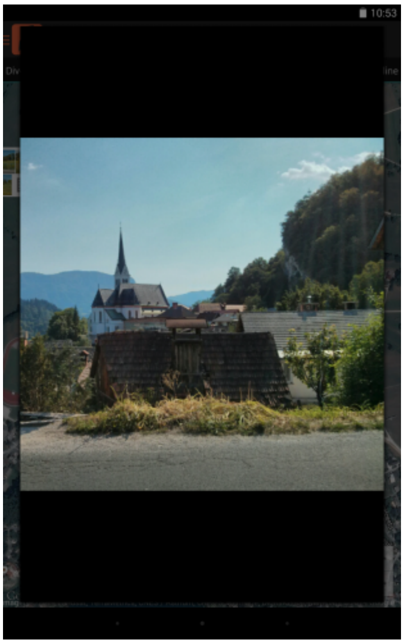

Figure 4.13: The *Element* mode.

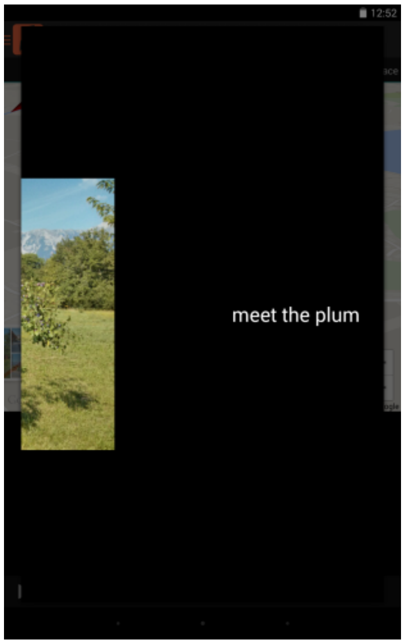

Figure 4.14: Swiping in the *Element* mode.

allowed to reason about the communication between the modes very easily. In particular, every mode's user interface was associated with its own actor, and an additional *Coordinator* actor was used.

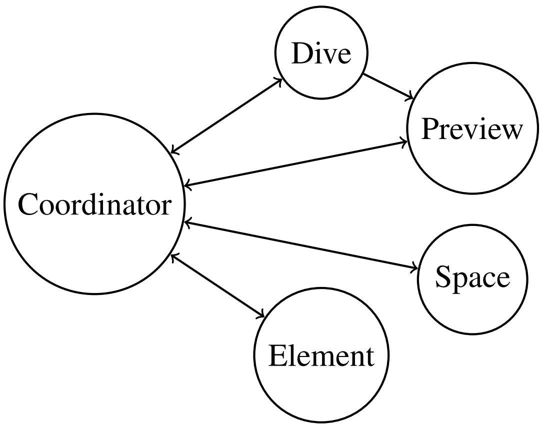

Figure 4.15: An actor system for viewing stories.

All modes were linked through the *element* domain, maintaining the focus on a particular story element. As an element is selected in any mode (through dragging the animation slider in the *Dive* mode, or swiping in the *Element* mode, etc), the selection is disseminated between all the other modes. Implementation-wise this means that any mode can notify the *Coordinator* about changing the focus by sending it a message `UpdateFocus(index)`, where `index` is the index of the new element in focus. The *Coordinator* will then resend the message to all other connected interfaces, obtaining synchronization.

In addition, the Dive mode map and its preview window were linked both in the element and magnification domains: as the map scales, the preview will display the elements with different granularity. If a cluster of three images and a text note is in focus on the map, the preview is going to visualize this exact cluster through a custom animation. This is achieved by exchanging an additional type of message between these two actors: `UpdateScale(scale)`, where `scale` is the smallest allowed distance between two markers in radians.

Finally, a connection between maps, timelines and grids was established: clicking on a story element in any viewing mode will trigger the *Element* viewing mode with that element in focus. Clicking on a map cluster will display the element selection dialog, allowing to select a particular element from that cluster.

4.2 Macroid

The Android platform provides a framework where graphic user interface components (called "layouts") are authored and stored in the form of XML files. However easy to use, the file-based approach imposes limits on composition of layouts from smaller pieces, as each of them needs to reside in its own file — a luxury, given the number of such pieces in a complex application and the lack of any namespacing mechanism. Even worse, to account for different screen orientations and densities, declarations are often duplicated with minimal changes. At the same time, the alternative Java API is verbose and does not allow for concise layout definitions.

Proper use of threads is another headache for the developers, as UI code and UI code only should be run on the UI thread to avoid concurrency problems and maintain a "snappy" feel of the interface. Finally, there is no simple approach towards the orchestration of asynchronous events and animations, in which the widgets engage.

We have developed *Macroid* — a functional UI domain-specific language for the Android mobile platform. *Macroid* is written in Scala and available as a library, which can be used in any Scala Android project. Complete source code and the accompanying documentation can be found at `https://github.com/macroid/macroid` and `https://macroid.github.io` respectively. In this section we would like to briefly outline the main features of *Macroid*. In particular, our contributions are as follows:

- We addressed the concurrency issues through the use of *UI actions* (section 4.2.1).
- Taking advantage of multiple Scala features, we introduced an elegant DSL for UI layout composition (section 4.2.2).

- We devised a functional façade for an object-oriented UI model to easily compose widget styles and behaviors (section 4.2.3).
- We simplified the definition of reusable adaptive components with a light-weight *media query* DSL (section 4.2.4).
- We further extended our UI model with asynchronous workflows and elements of functional reactive programming (sections 4.2.5, 4.2.6).

4.2.1 UI Actions

An issue that underpins UI development is that of correct threading. Android documentation provides two golden rules for dealing with the UI thread[8]:

- *Do not block the UI thread;*
- *Do not access the Android UI toolkit from outside the UI thread.*

In the past, similar problems have been dealt with through the use of effect systems [17] and/or monads [43][9]. We propose a `Ui` monad to contain UI effects, along the lines of the `IO` monad. We say that a value of type `Ui[X]` is a *UI action* and equip such values with a method `run` that sends the action to the UI thread for execution. There are three benefits to this approach: 1) UI actions demarcate parts of the code that need to be executed on the UI thread, facilitating program comprehension. 2) We can ensure that the UI code is always run on the correct thread. 3) When submitting several UI commands from a background thread, we can combine them in an atomic operation.

An example of creating and running a UI action is as follows:

```
val action = Ui {
  textView.setText("caption")
}

action.run
```

UI actions can be composed in a number of ways and are used as return types throughout *Macroid*.

[8]See `http://developer.android.com/guide/components/processes-and-threads.html#Threads`

[9]`https://github.com/raimohanska/ui-thread-as-monad`

4.2.2 Layout DSL

Android XML layouts consist of declarations of widget hierarchies and their properties. The lack of a namespacing mechanism for the corresponding XML files (e.g. introducing sub-folders) results in global namespace pollution, as the number of layouts increases greatly for complex applications. In addition, only select properties can be defined in XML, forcing the developer to write “glue code” in Java or Scala to attach event handlers and behaviors.

In order to overcome the above limitations, we would like to be able to declare them with ordinary Scala code, taking advantage of existing Scala features, such as values, methods, objects and traits, to improve layout modularity.

Unfortunately, the standard Android API is rather verbose for this task. First, a context parameter has to be passed to each widget; second, class constructors produce unnecessary visual noise:

```
new LinearLayout(ctx) {
  this.addView(new Button(ctx))
  this.addView(new TextView(ctx))
}
```

We thus turned to the metaprogramming techniques introduced by Scala macros [6]. *Def macros*, which we employ below, are Scala functions operating on abstract syntax trees of Scala code, and are called by the compiler. Essentially, they allow to specify code transformations, that the end-user’s program will undergo. Utilizing macros, we achieved the following transformation:

```
widget[Button] ⟶ Ui(new Button(ctx))
```

In a similar manner, a macro for parent widgets, such as `LinearLayout`, was defined, allowing us to rewrite the layout example above more elegantly:

```
layout[LinearLayout](
  widget[Button],
  widget[TextView]
)
```

The presented DSL surpasses XML layouts in terms of composability: for example, Scala’s objects and values can be used to declare smaller layout pieces, while objects allow to group these pieces into modules:

```
object Layouts {
  val layout1 = widget[Button]
  val layout2 = widget[TextView]
}
```

```
val layout3 =
  layout[LinearLayout](
    Layouts.layout1,
    Layouts.layout2
  )
```

It is worth mentioning that XML layouts still offer some advantages over our approach, as they can be edited visually and provide an immediate feedback without the need for recompilation. However we argue that the value of visual editing is reduced greatly when custom (i.e. unsupported by the editor) widgets are utilized, or when parts of the layout are created programmatically. Likewise, immediate feedback is only available when certain properties are changed, excluding e.g. any event handlers or behaviors attached to the widgets.

4.2.3 Tweaks

Android widget property configurations, declared in XML layouts, do not offer any means of composition. To address this, we introduced the notion of *tweak* — a side-effecting change of one or several widget properties. More precisely, a tweak is defined by a function $Widget \Rightarrow Unit$, for example:

```
def text(str: String): Tweak[TextView] =
  Tweak { widget: TextView ⇒
    widget.setText(str)
  }
```

Essentially, we embraced UI side-effects, at the same time providing a high-level functional façade for the underlying object-oriented model, allowing to manipulate widget properties in a number of interesting ways, which we discuss below.

In conjunction with Scala's values, methods, objects and traits, tweak composition, denoted by the '+' operator, serves as a powerful tool for defining custom widget styles and behaviors, for example:

```
def largeText(str: String) =
  text(str) +
  TextTweaks.large
```

In order to apply (a series of) tweaks, the *tweaking operator* '<~' is used, producing a UI action:

```
val tweakedTextView: Ui[TextView] =
  textView <~ text("caption") <~ show
```

More generally, a tweaking operation w <~ t for `w: W` and `t: T` is defined and produces a value of type `Ui[R]` if a corresponding instance of multi-parameter type class `CanTweak` is found in the implicit scope (see [33] for introduction to type classes and their usage in Scala):

```
trait CanTweak[W, T, R] {
  def tweak(w: W, t: T): Ui[R]
}
```

This design, reminiscent of collections in the Scala standard library, allowed us to extend tweaking to support various type constructors, such as `List` and `Option`. We further expand on this by considering *futures* and *event streams* in section 4.2.6. To illustrate the usefulness of this generalization, we are able to tweak widgets in batches:

```
List(button1, button2) <~ maybeHide
```

4.2.4 Media Queries

The previous section brings us to the concept of *media queries*, borrowed from the CSS feature of the same name[10]. Packing not much more than a `Boolean` value and a handful of convenient operators, media queries are nevertheless much more flexible than Android's XML approach, which requires to duplicate layouts and save them into folders corresponding to different screen orientations and densities.

Examples of media queries are: `widerThan(100 dp)` (dp stands for *device pixels*), `landscape`. We defined a '?' operator on media queries, the expression `query ? x`, `x: X` having the following semantics: *a*) the resulting type is `Option[X]`; *b*) if the `query` condition holds, the result is `Some(x)`; *c*) else, the result is `None`. Additionally, a '|' operator is provided for *options* as an alias for `orElse` and `getOrElse`.

With the above definitions, we are able to compose adaptive tweaks and layouts:

```
val adaptiveSize: Tweak[TextView] =
  widerThan(1000 dp) ? TextTweaks.large |
  narrowerThan(200 dp) ? TextTweaks.small |
  TextTweaks.medium

textView <~ adaptiveSize
```

Effectively, by carrying adaptation out at the smallest level — tweaks and tiny layout pieces — we enabled a more fine-grained approach, abolishing unnecessary code duplication.

[10] `http://www.w3.org/TR/css3-mediaqueries/`

4.2.5 Snails

Designing visually appealing interfaces often requires incorporating animation (e.g. fade-in, fade-out). However, imperative and callback-based programming techniques are poorly suited for defining complex animation workflows. Dataflow programming [21] provides a better ground, emphasizing declarative approach and modeling flow graphs rather than steps of execution.

We devised a dataflow approach to animations, based on *snails* — side-effecting asynchronous changes of widget properties. A snail is conceptually similar to a tweak and is defined by a function $Widget \Rightarrow Future[Unit]$. The return value of type `Future[Unit]` is used to determine when the property change is finished. `fadeIn` is an example of a snail that runs the fade-in animation.

The *"snailing" operator* '<~~' is used to apply (a series of) snails, constructing entire animation workflows, and it can be mixed with tweaking (the `delay` snail used in the following example does not change any widget properties, but rather introduces a delay):

```
textView <~~
  fadeIn(300) <~
  text("caption") <~~
  delay(3000) <~~
  fadeOut(300)
```

The semantics of applying a snail are as follows: 1) the animation, denoted by the snail, is run; 2) once the animation is finished, the execution proceeds to the right of the chain. Thread management is done automatically, without any blocking.

Snails can be combined with each other or with tweaks:

```
val wink =
  fadeIn(300) ++ fadeOut(300)

def textAndFade(str: String) =
  text(str) ++ fadeIn(300)

val fadeAndDisappear =
  fadeOut(300) + hide
```

By combining the snailing UI actions, more elaborate effects can be achieved:

```
(myProgressBar <~~ fadeOut(400)) ~~
(myTextView <~~ wink) ~~
(myOtherTextView <~ text("Hello"))
```

4.2.6 Futures and EventStreams

Picking up from section 4.2.3, where we defined an extendable way of tweaking widgets, we would like to explore how tweaking can be connected with *futures* and *event streams* to enable asynchronous and functional reactive programming [29] approaches.

First, if we consider a value of type `Future[String]` and a text box, the result of the future could be assigned to the text box just in one line (actual assignment will happen *in the future*, when the value is ready):

```
val futureCaption: Future[String] =
  Future {
    ...
    "Message"
  }

textView <~ futureCaption.map(text)
```

Functional reactive programming, mentioned previously, is based upon introducing time-varying entities and establishing connections between them, causing propagation of changes.

In a manner similar to the above example, we can utilize one of the available Scala implementations of reactive (time-varying) variables, e.g. *scala.rx*[11], to achieve reactive change propagation:

```
// create a reactive variable
val caption = Var("Hello")

// set text to "Hello"
textView <~ caption.map(text)

// text automatically updates to "Goodbye"
caption.update("Goodbye")
```

Another example is the use of event streams, provided e.g. by *scala.frp*[12]:

```
// create an event source
val clicks = EventSource[Unit]()

// setup a button to fire events
button <~ On.click(Ui(clicks.fire(())))

// a more useful event stream
val randomInts = clicks.map { _ ⇒
```

[11] `https://github.com/lihaoyi/scala.rx`
[12] `https://github.com/dylemma/scala.frp`

```
  scala.util.Random.nextInt().toString
}

// the caption updates with each click
textView <~ randomInts.map(text)
```

While still requiring a certain amount of boilerplate, the above approaches greatly simplify the development of reactive interfaces and asynchronous workflows by utilizing the *tweaking* abstraction and providing automatic thread management.

4.2.7 Related Work

Over the years, several functional UI frameworks have been devised, mostly for the Haskell programming language, for example [39, 11, 35]. A comprehensive survey of these works can be found in [41]. Recently the Elm language [12] was designed for reactive web application frontends. As Peterson *et al.* [35] demonstrates, UI models, which are based on functional reactive programming principles [29] from the ground up, possess an inherent mismatch to the object-oriented UI toolkits, such as the ones found on mobile platforms. Moreover, none of the frameworks listed above runs natively on any current mobile platform.

Odersky and Maier [31] describes several functional reactive techniques (FRP) applied to the *scala-swing*[13] UI toolkit. Being independent of the concrete FRP implementation, *Macroid* can be used with this or similar systems.

A few FRP libraries are available for mobile platforms, including *RxJava-Android*[14], *ReactiveCocoa*[15], however they solely focus on the functional reactive aspect, leaving layout composition out.

Finally, *Scaloid*[16] is a Scala framework, aiming to simplify Android application development. Mostly consisting of wrapper classes and implicit conversions, it favors only a particular set of widgets, lacks the compositional advantages of *Macroid*'s tweaks and media queries, and does not offer any dataflow capabilities.

4.2.8 Conclusions and Future Work

We have briefly introduced *Macroid*, a functional UI DSL for Android, and argued that it enriches mobile interface development with new degrees of flexibility, composability and

[13] https://github.com/scala/scala-swing
[14] https://github.com/Netflix/RxJava/tree/master/rxjava-contrib/rxjava-android
[15] https://github.com/ReactiveCocoa/ReactiveCocoa
[16] https://github.com/scala/scaloid

expression. We have also demonstrated how particular Scala features played a great role in the advancement of our DSL.

As of this writing, *Macroid* has been presented at two major public events:

- Codebits 2014, Lisbon, Portugal, April 2014 (`https://codebits.eu/intra/s/proposal/455`).
- ScalaDays 2014, Berlin, Germany, June 2014 (`https://www.parleys.com/play/53a7d2cfe4b0543940d9e564`).

We have also developed a few smaller libraries under the *Macroid* umbrella: *Macroid-Akka-Fragments* provides glue code for using Akka on Android; *Macroid-Viewable* defines type classes for converting data structures to Android layouts. Their homepages can be found at `http://macroid.github.io/Related.html`.

In the future, we wish to focus more on the functional reactive aspects of *Macroid*, reducing the currently necessary event-handling boilerplate.

Another alluring research area lies in auto-generating interfaces for data structures, employing implicit macros for type class instance materialization [6]. Similar work has been done (using Template Haskell) in Polak and Jarosz [36].

Finally, Gajos *et al.* [15] demonstrates how interfaces could be automatically adapted to account for certain device types, input methods and disabilities of the users. In the same vein, we wish to investigate whether tools for such adaptation could be integrated into *Macroid*.

Chapter 5

Evaluation

> *In this chapter we describe the study we have conducted in order to gain insight into how people record stories in the wild, using our route storytelling approach, as well as how third party viewers perceive the resulting stories.*

5.1 Research Questions

We aimed to answer four research questions, the first three of which were related:

Does presenting suggestions during authoring...

- *RQ1*: ...improve authoring experience?
- *RQ2*: ...increase authors' satisfaction with their stories?
- *RQ3*: ...improve viewing experience of the third parties?

The fourth question was concerned with the individuality of the stories:

- *RQ4*: What are the authors' and the viewers' perspectives on stories' individuality, in the presence of suggestions?

Finally, as in our preliminary study, we wished to better understand how people interacted with our storytelling platform, what types of media were used the most, and what our participants considered to be a good story.

5.2 Methodology

5.2.1 Phase 1

In this phase we sought to answer *RQ1* and *RQ2*. We conducted an in-the-wild experiment with the participants using our mobile application. We randomly divided the participants into two equally sized groups. In case of the experimental group (*Group B*), the application presented third party content in the form of suggestions, which could be added to the stories during authoring (and functioned identically otherwise). We collected and compared sets of stories recorded by each participant; we measured participants' authoring experience and satisfaction with per-story Likert scale questions, as well as a final questionnaire.

Apparatus

We have built a website (`http://routestory.net`) with a trifold purpose: *a*) to engage people and familiarize them with route storytelling and our study; *b*) to register future participants; *c*) to divide the participants into groups in an automatic (and randomized) fashion.

Our application was used by all participants.

A set of Likert scale (1 to 5) questions was used to measure authors' satisfaction and viewing experience. Here we aimed to match the measurements with those obtained from third parties in the *Phase 3*. At the same time, we ensured that author's satisfaction with a story and its estimated value for third parties were clearly separated, since a story can be satisfying without being intended for sharing. We also probed for media quality to analyze where the suggestions stood. The questions were presented at the end of the study, so that the participants could review and reflect on the stories they were rating and submitting. The questions were worded as follows:

- *How satisfied are you with this story?*
- *How would you rate the overall quality of the media (photos, sounds, etc) in this story?*
- *How interesting would this story be to someone else?*

At the end of the study we used the following questionnaire to establish "author profiles" of our participants:

1. *When walking or traveling, I normally…*
    - *Do not take any photos and/or notes*
    - *Do not take any photos and/or notes myself, but obtain them from fellow travelers*
    - *Take a few photos and/or notes*
    - *Take a large amount of photos and/or notes*

2. *When I take photos and/or notes, I want to share them with other people* (on a scale from 1 to 5; 5 = *"agree"*)

3. *When I take photos and/or notes, I want to keep them for memories* (on a scale from 1 to 5; 5 = *"agree"*)

Lastly, taking the opportunity to assess our storytelling system, we handed out two standard questionnaires:

- SUS [4] and
- 10-item short version of AttrakDiff2 [19]

Participants

We recruited 25 participants through the word of mouth, mailing lists and social networking, of which 14 have completed the study (6 from *Group B*). Participants were not remunerated.

Procedure

Each future participant had visited our web page (`http://routestory.net`), familiarizing themselves with the concept of route storytelling, the outline of our study procedure and our privacy policy (see section A of the appendix), and registered for the study by supplying their e-mail in the input form. We subsequently authorized them for downloading the application, which was done through the Google Play Store beta testing program.

When the application was launched for the first time, it prompted for the user's e-mail account and sent a registration request to our server. The server randomly assigned the e-mail to either *Group A* (control group) or *Group B* (experimental group), communicating the decision back to the application. The application thus enabled or disabled suggestions depending on the response.

If the connection to the server could not be established, it was retried later, however the application would assume suggestions to be disabled for the time being. The rationale for this approach is as follows: *a*) given the connection failure, it would be unlikely for the application to successfully fetch the suggestions anyway; *b*) once the user has been given suggestions, it would be a mistake to assign them to the control group later on; *c*) if suggestions are only enabled by the time the user records some (but not all) stories, we could still assign them to the experiment group based on the remaining stories. The application also kept track of whether any suggestions were actually shown.

The participants were instructed to use the application to record stories “in the wild” during the course of 5 days (starting from the time they acquired the application).

As the 5-day period ended, the participants were required to select a minimum of five stories, at least 5 minutes long each, and to send them to us using the application. During this process they were also asked to rate each of these stories, specifying their satisfaction with the story, its value for a third party viewer, and the quality of media employed.

Finally, upon receiving the stories we distributed the questionnaires and gathered the responses.

Measures

For each story we have obtained the following, as rated by the authors:

1. author’s satisfaction;
2. media quality;
3. interest for the third party viewers.

We also extracted the following data:

4. density of elements (number of elements per second);
5. density of suggested elements, if any were available during the recording.

We studied the correlation of measures 1–3 with one another (pairwise), as well as their correlation with measures 4 and 5 (pairwise), first using all the stories as our dataset and then between groups *A* and *B*.

We also compared the results obtained from the user experience questionnaires between the groups.

5.2.2 Phase 2

In this phase we aimed to answer *RQ1* (*Does presenting suggestions during authoring improve authoring experience?*) and *RQ2* (*Does presenting suggestions during authoring increase authors’ satisfaction with their stories?*) with more qualitative data, as well as to touch upon *RQ4* (*What are the authors’ and the viewers’ perspectives on stories’ individuality, in the presence of suggestions?*). We picked several participants from the *Phase 1*, equally representing both *Group A* and *Group B*, and invited them for an interview, discussing their authoring experience and satisfaction with their stories.

Apparatus

The following semi-structured interview script was used:

1. *What do you think about the concept of route storytelling?*
2. *Would you say that route storytelling requires significant authoring effort?*
3. *What in your opinion makes for a satisfying story?*
4. *Did you feel there was a compromise between authoring effort and satisfaction with the result?*
5. *What do you think about the suggestions?*[1] *Do they reduce the authoring effort? Do they make stories less personal?*

Participants

From the *Phase 1* participants, 6 people were selected, based on the content of their stories and their availability for an interview. These people equally represented both *Group A* and *Group B* from the *Phase 1*.

Procedure

Upon analyzing the collected stories, we selected 6 participants for this phase. In particular, the three people we selected from *Group B* differed in the amount of suggested elements employed in their stories: two have used them to a varying degree, and one has not used any suggestions (although has received them).

The authoring profiles of our interviewees are provided below.

Participant	Group	Typical amount of media	Media intended for keeping memories	Media intended for sharing	Usage of suggestions
P1	B	Small	3/5	4/5	0.08
P2	A	Medium	5/5	2/5	—
P3	B	Medium	5/5	3/5	0.15
P4	A	Large	5/5	5/5	—
P5	A	?	?	?	—
P6	B	Medium	4/5	4/5	—

Columns 3-5 should be interpreted as answers to the following:

- *When walking or traveling, I normally... [capture the following amount of media]*

[1] For *Group A* we provided an explanation of the term and briefly described our implementation.

- *When I take photos and/or notes, I want to share them with other people*
  (on a scale from 1 to 5; 5 = *"agree"*)
- *When I take photos and/or notes, I want to keep them for memories*
  (on a scale from 1 to 5; 5 = *"agree"*)

Column 6 represents our measurements of suggestions incorporated per minute of a story, averaged by all of the author's stories.

We conducted semi-structured interviews either in person or via Skype calls, typing in summaries as we spoke (translating to English if necessary). The summaries were shown to the interviewees to verify that they were accurate. Two of the participants from *Group B* (participants 4 and 5) were a couple and were interviewed together.

5.2.3 Phase 3

In this phase we wished to answer the *RQ3* (*Does presenting suggestions during authoring improve viewing experience of the third parties?*) and *RQ4* (*What are the authors' and the viewers' perspectives on stories' individuality, in the presence of suggestions?*). We recruited additional participants and presented them with a subset of the stories recorded in the *Phase 1*. We discussed and measured their viewing experience.

Apparatus

Based on several criteria (see below), we picked 16 stories recorded in the *Phase 1* (8 from each group). We used our application to display the stories on a 7-inch screen tablet to allow seamless exploration.

As in the *Phase 1*, a set of Likert scale (1 to 5) questions was used to measure participants' viewing experience. These questions reciprocated the aforementioned ones:

- *How would you rate the overall quality of the media (photos, sounds, etc) in this story?*
- *How interesting is this story?*

Participants

We recruited 8 participants (neither of whom has participated in *Phase 1* — and consequently *Phase 2*) by intercepting them in the university campus area. Participants were not remunerated.

Procedure

Within both *Group A* and *Group B*, we have ranged all the collected stories using the following criteria (in order of precedence):

1. author's rating of the story's interest to the third party viewers (higher is better);
2. density of elements, i.e. amount of elements divided by story length (higher is better).

We then picked the top 8 stories in each group to form a list of 16 most compelling stories. In case of *Group B*, only the stories that actually featured suggested elements were used.

Our 8 participants were split into 2 groups of four, each group undergoing the same procedure with the same set of stories, for redundancy of the obtained ratings. We randomly preassigned our 16 stories to each group of 4 participants, so that each person would see 4 stories, two of each kind.

After viewing each story, every participant rated it using our Likert scales. To calibrate the ratings and give an introduction to the concept, an extra story (the same for all participants) was shown before the other four. Its ratings were discarded.

Throughout the study, we encouraged every participant to share their thoughts on the concept of route storytelling, our application, and the stories they have seen, as we took notes. We also explained the idea of suggested elements and discussed its implications for stories' individuality, and the degree to which the participants expected the stories to be personal.

Measures

For each story we have obtained the following, as rated by the viewers:

1. media quality;
2. degree of interest.

We considered the ratings of the same story, given by two participants, to be in agreement if they were no more than 1 point apart. We averaged the pairs of satisfying ratings and tested them for correlation with the measures obtained in the *Phase 1*, overall and between the groups *A* and *B*.

5.3 Results

5.3.1 Phases 1 and 2

Route storytelling

Most interviewees sympathized with the concept of route storytelling, albeit acknowledging that the stories may take effort not only to produce — as we expected, — but also to consume.

> *For example, in Instagram not many people actually watch videos. So I'm not sure people will watch a big route story.* [...] *I don't think anyone would ever be interested in the five stories I recorded during the study. So* [route storytelling] *is not for the daily life. More suited for big walks or travels.* — commented *P4*.

The idea of route storytelling not being suitable for everyday life was shared by all interviewees, most concisely expressed by *P2*: *"I'm lazy to do it. Maybe that's why it's not so popular."*

Media usage

By the end of the study, we have received 58 stories from 14 participants. As seen in the figure 5.1, images were the most frequently used type of media; we found no statistically significant difference between the groups with regards to the amount of images used.

Authoring experience

Our system has obtained a SUS rating of 84.8 (sd=6.0). Favorable scores have been received in the AttrackDiff2 questionnaire, as demonstrated in the figure 5.2.

We have not found any significant differences between the two groups.

Authoring effort

Most interviewees agreed that some effort pertains to the authoring process, noticing that it is more true if a good story is desired. Time was listed by *P5* as another important resource, required for the process. *P6* concurred:

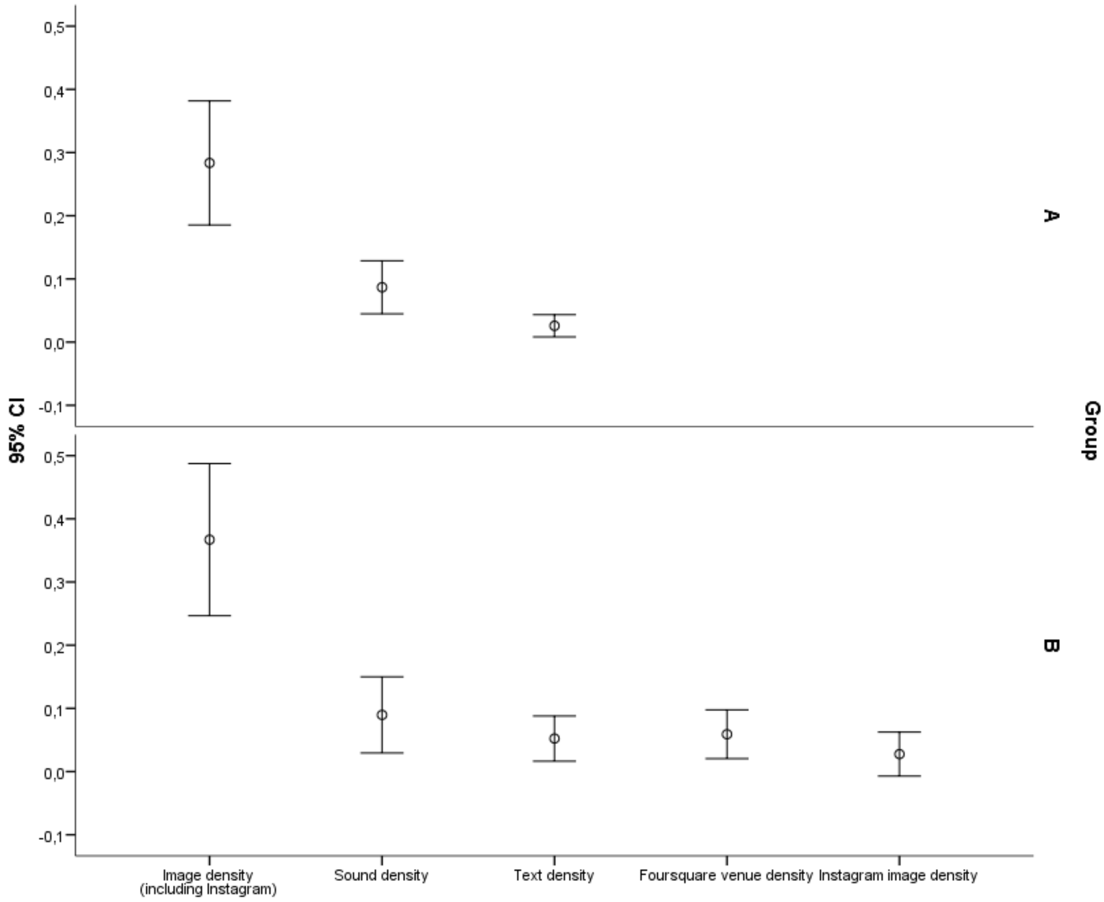

Figure 5.1: Media usage statistics (in elements per minute).

> *In one story we were doing geocaching. While my brother and his girlfriend were searching for the cache, I lost some time recording the story. As a result, I haven't even witnessed the moment they had found it.*

P3 noted that the effort was reduced by automatic recording of ambient sound, as well as suggestions.

Authoring satisfaction

We observed two distinct aspects, which, according to our interviewees, contributed to the author's satisfaction.

The first aspect deals with the intrinsic qualities of the story. Most strongly advocated by *P3* (whose profile demonstrated a preference for keeping memories over sharing media), this includes the quality of the media, the interestingness of the route and the locations themselves, the meaning and history of the places visited, even the distance covered:

> *The longer the route, the more interesting things (photos, places) it has. It is*

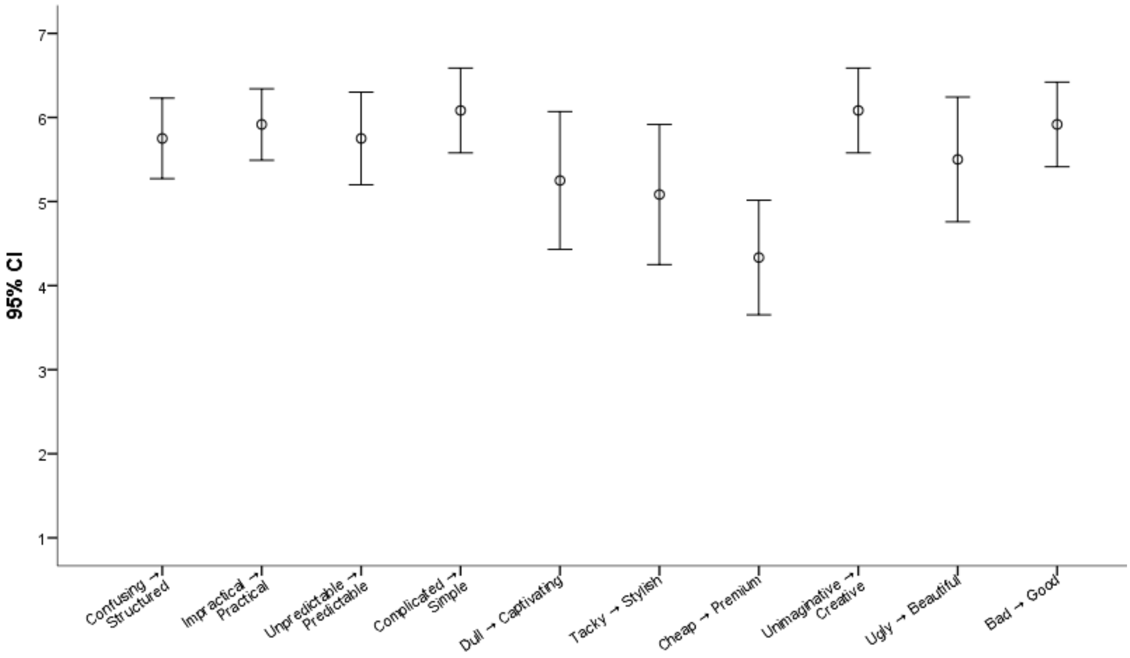

Figure 5.2: Results from the AttrackDiff2 questionnaire.

> *more spread out. If the route is very short, I have all the pictures almost in the same place, and the reproduction is not that pleasant. The pictures are very close, so it's like I've always been at the same place. More of a place story than a route story.*

Another aspect, however, is that of the story's authenticity and its connection to the audience. *P4* (who was interested in sharing media more than other participants) commented:

> *The media should be good, but still it's very important that people like the story itself.* [...] *It's little things, unique details and an atmosphere of human interactions that make the Story, not just some text and loads of photos.*

P5 asserted that a lot of popular content on the Internet is not necessarily of good quality:

> *Sometimes it happens that you make a crappy story, and people really like it. And sometimes you make a good story, but nobody cares about it.*

Thus it is possible for a story to score high in one aspect, but not the other. The resulting satisfaction then depends on the personal preference for either of them.

The quantitative data is in line with the above results: we have found a statistically significant correlation between author's satisfaction and media quality ($\rho(50) = 0.748, P = 0.00$), a borderline significant correlation between author's satisfaction and story's element density ($\rho(50) = 0.295, P = 0.03$), and a significant correlation between author's satisfaction and story's interestingness (as rated by the author) ($\rho(50) = 0.539, P = 0.00$).

Effort vs satisfaction

Most interviewees identified a correlation between the effort put into the story and their satisfaction with the outcome. Again, *P2* put it most concisely: *"The more effort you put, the better the story. That's exactly why I rated my stories so low — I did not put much effort."*

The relation however was not perceived as linear by all of our interviewees. Some of them considered the effort to have a threshold after which the story does not improve much or even degrades: *"Maybe there is a limit to improving the story. Maybe if you put too much information, the story becomes very confusing."* — commented *P6*.

Suggestions

Overall, interviewees from *Group A*, who have not seen our implementation of the concept, at first seemed to be less enthusiastic about it:

> — *Suggestions are not an important feature. I will not benefit much from them. — P2*

> — *I resist this idea strongly. — P4*

> — *It very much depends on the implementation. — P5*

As we uncovered more details and use-cases behind the idea, the reaction graduated towards a more positive one. *"Integrating your story with others' opinions, like from Foursquare, would be cool."* — reflected *P4*.

Interviewees from *Group B* tended to focus more on particular ways to improve the feature and make it more useful, which we discuss later in this section. Still, there was an indication of suggestions being beneficial in their then-current form:

> — *Some suggestions were helpful.* [. . .] *Once I heard a plane flying, and funnily enough, there was a picture with a plane among the suggestions, so I added it. — P1*

> — *I like the suggestions. Very good feature. — P3*

Several participants from *Group B*, who were not interviewed, nevertheless shared their feedback, at times to striking contrast:

> — *I loved it.*

— *The pictures from* [Instagram]*, I don't understand why they are there as it seems a bit pointless. If you want to create your own story, why would you want to add a ton of pictures from other people from* [Instagram]*? Maybe it would be nice to have the option to do that, but to see a ton of pictures is a bit too much in my opinion.*

The latter brings us to the first concern, a thorough work on which might result in a more positive reaction towards suggestions, namely their

Relevance. With contemporary personal media production trends in mind, many of the suggested images included portraits or self-portraits:

— *I did not like being "spammed" with photos of people I didn't know. — P1*

— *There could be a better filter to avoid faces etc. It would make* [suggestions] *more interesting. I would use them more. — P3*

— *I didn't include the suggestions, because they were not relevant. For example, I was showing where the library of my city was, and I saw pictures of some people. I was more focused on my story, so I didn't pay much attention to suggestions after that.* [...] *If the suggestions were more relevant, it would be a useful feature. — P6*

A more sophisticated algorithm, such as [42], could be used to sift the imagery by taking advantage of its metadata. Another approach would be to simply use a less persona-oriented web service, such as Flickr or Panoramio[2]. Although we did implement a Flickr client, no media was found at locations where we did our own tests, and we suspect that our participants had not received Flickr suggestions either.

Foursquare venues, while perceived as generally more relevant, uncovered another problem: as the number of suggestions was fixed, the venue being sought was not always on the list, outshadowed by the less prominent ones. This leads us to believe that a ranking scheme should be properly configured for such services, as sorting by distance from the current location does not always produce the desired results. On the other hand, increasing the number of suggestions displayed or allowing to browse and search for them interactively may introduce additional authoring effort, as we discuss in the next paragraphs.

Effort and satisfaction. Pinpointing the role of suggestions in the trade-off between authoring effort and satisfaction, *P1* explained:

Suggestions potentially improve the story, but don't reduce the effort. I mean, they can, but normally they don't, because you still need to look for a good

[2] `http://www.panoramio.com/`

suggestion, and some of them might not even be relevant.

It seems that the effortlessness of adding a suggested element is formed by two factors:

- *The expressive power of the element.* For example, in case of Foursquare venues a big amount of media is added at once, including visitors' photos, comments, etc.
- *The ease with which the element can be found.* As noticed by *P3*,

  > *If you are in front of a monument, most of the pictures will be of that monument, so it's easier to find the right one, and the effort is reduced. In other places, there is no attraction there, so* [...] *it takes more effort to find the one that is interesting.*

Overall, using suggestions did not seem to directly increase authors' satisfaction with the stories, and we have not found any statistically significant correlation in the quantitative data. Interestingly, *P2* hypothesized about the contrary: *"Maybe I will be less satisfied because someone has taken a better photo!"*

Individuality of the stories encompassing suggestions turned out to be perhaps the most controversial subject. *P4* commented:

> *Take the Eiffel tower. A lot of pictures have been made of it. But many of my friends (including myself) still take their own pictures. Because there is something personal to it: you were there, it's your experience. You put your own vision into it. If you take someone else's media, then your experience, your authenticity is lost.*

Still, most interviewees agreed that not every story needs to be personal. For example, when showcasing a certain place, be it a historic monument or a remarkable scenery, summoning the best media, external or not, might be welcome.

As we expected, the proportion of third party elements was stated to play a big role in the overall impression. In practice, for stories that included suggestions it varied greatly from 0.06 to 0.67 elements per minute ($mean = 0.17, sd = 0.17$). More importantly, *P4* expressed the need to be in full control of the amount of external data in the story, thus supporting our suggestion-based approach rather than *automatic* enrichment or even generation of stories, found in several works, e.g. [42].

It also became apparent that elements of different types can possess or defy different amounts of story's individuality. In particular, photographs were mostly perceived as being extremely personal. With other media, however, the line is blurred. While interviewing *P3*, we discussed the individuality of several story element types more elaborately:

> [Information on current weather] *is personal, because it is experienced by the*

> *author. On the other hand* [if it is a suggestion coming from a web service], *maybe it's not something the author wanted to explicitly add to the story in the first place.* [A fact from Wikipedia] *is not personal because the story's author did not write it, but it doesn't mean that they didn't feel that way. So it was probably their opinion as well that they wanted to add, that they felt was important to be in the story. Definitely less personal than a text note, but doesn't necessarily make the story less personal.*

This insight and the feedback from other interviewees lead us to propose the following classification, in order of decreasing individuality (examples in parentheses):

1. Personal content (photos, notes).
2. Logged data (ambient sound, SenseCam [20] photos).
3. Automatically gathered data (current weather).
4. Crowdsourced data (Wikipedia facts, Foursquare venues).
5. Others' personal content (Instagram and Flickr photos).

Repurposing. Two of our interviewees, namely *P2* and *P4*, proposed to use suggestions as route and sightseeing recommendations, rather than potential additions to the story:

> *It's better to have suggestions on where to go. Something like "If you turn right now, you will see* ***this****!"*

This concept has already been explored by Mitchell and Chuah in [27].

5.3.2 Phase 3

Ratings

Out of 16 stories, only 4 had their media quality ratings in disagreement. For the interestingness rating, the number of stories in disagreement was 5. We have not found any statistically significant differences between the ratings of stories from *Group A* and *Group B*, or any correlations of viewers' ratings with authors' ratings. However, there was a strong correlation between the viewers' rating of story's interestingness and its element density ($\rho(9) = 0.830, P = 0.02$), with weaker correlations between the former and image density ($\rho(9) = 0.734, P = 0.01$) and Foursquare venue density ($\rho(4) = 0.812, P = 0.05$).

Suggestions

Many viewers compared the concept of suggestions with crowdsourcing, emphasizing that the additional data provided a broader perspective on the story:

> — *Incorporating Foursquare venues is interesting, because the story is no longer limited to just the author, and provides an opportunity for comparison. — P1*

> — *Instagram integration is helpful, because you can find pictures of better quality than those of your own. It's fun to have many perspectives. — P6*

Individuality of the stories seemed to concern the viewers much less than it was the case with the authors. Only two participants expressed a strong preference for personal stories; still, both of them felt that there were valid reasons to employ external media:

> — *More personal stories are better. Pictures on the Internet are "old": for example, sometimes you go to a place and there is a fair, which you wouldn't find on a picture taken previously. But if the place has not changed, sure, why not use an existing one. — P2*

> — *Original content is better. External media should act as a backup for when you can't take your own picture, for example, because of weather conditions. — P5*

Overall, the perceived degree of individuality, associated with various story elements (although rarely discussed in detail), was in agreement with the five categories we proposed above. In particular, ambient sound was considered more personal than the suggested elements, followed by Foursquare venues and ultimately Instagram photos.

5.4 Summary

In the present study we set out to investigate whether the use of suggestions can reduce the authoring effort or improve viewing experience. Below we are summarizing our findings with respect to the research questions.

Does presenting suggestions during authoring improve authoring experience? We have not found any statistically significant differences between the experimental and control groups. However, qualitative data obtained through semi-structured interviews suggests that suggestions might indeed reduce authoring effort, provided that they are relevant, readily available and possess sufficient expressive power.

Does presenting suggestions during authoring increase author's satisfaction with

their own stories? To this end, we have surfaced the criteria authors employ to determine their satisfaction with their own stories. As their desires are split between perfecting the intrinsic qualities of the stories and catering for the audience, a certain benefit of suggestions may be inferred from the rest of our results.

Does presenting suggestions during authoring improve viewing experience of the third parties? We have not found any statistically significant differences between the viewers' ratings of the stories produced by the control and experimental groups. However, we did find a significant correlation between the ratings of the story's interestingness and the proportion of elements of a particular type (Foursquare venues), which were incorporated through the mechanism of suggestions. Further, the third party viewers welcomed the idea of enriching stories by reusing the content from the Internet, as long as this content provided a broader perspective on the story or its locus in quo.

What are the authors' and the viewers' perspectives on stories' individuality, in the presence of suggestions? Through interviews with both authors and viewers, we have found that the former were concerned about losing the individuality of their stories by incorporating suggestions. Still, they agreed that by manually controlling the proportion of external story elements, the right balance might be achieved. As we mentioned above, the viewers were generally less strict and preferred more comprehensive, rather than more personal, stories.

Chapter 6

Conclusions

We have proposed and evaluated an approach for improving the authoring and viewing experiences in the area of route storytelling. The results suggest that our solution is viable, although can be further ameliorated.

In the course of the present work, we have contributed a storytelling system, a few open source libraries and described a framework suitable for evaluation of future storytelling systems.

6.1 Limitations

As flexible as our system is, we have not taken full advantage of the chapter-based story structure we had proposed, thus depriving ourselves of working with more complex stories. For example, making a pause during the recording could have been implemented through concatenation of two separate chapters (before and after the pause), yet our current implementation does not support pausing.

During the evaluation, a greater number of participants could perhaps have allowed us to obtain more quantitative results, however we underestimated the percentage of participants who were not going to complete the study in full, and received less data than we had hoped for.

6.2 Future work

A promising research direction is that of collaborative storytelling, as it exploits the fact the many experiences are shared. Fails *et al.* [14] demonstrated how collaboration and task splitting could improve the quality of the story. It may be also interesting to investigate a setting, where the authors are not located at the same place. For example, Lindley *et al.* [24] describe a case of spouses being intrigued to reflect on recordings of each other's day, as the husband went to work and the wife did the housework. Our system has been designed to allow extension to collaborative scenarios and could provide the ground for new research on combining lifelogging, crowdsourcing and collaborative authoring.

Appendix A

User agreement

Every participant of the *Phase 1* of our main study has agreed to our privacy policy and study procedures, which were described on our website (`http://routestory.net/eula`) as follows:

> Thank you for your interest in RouteStory!
>
> We would be excited to have you as one of the very first users and are looking forward to your invaluable feedback.
>
> Here is an outline of our beta testing program, please read it carefully and let us know if any questions arise. The program is closely connected with a scientific study, directed at understanding how people share route experiences and at improving our route storytelling platform. There are a few terms to consider:
>
> 1. The application requires Android 4.0+ and relies on GPS and 3G connectivity.
> 2. The application will use the e-mail address from your Android account (which in most cases will be the same as the address you use for the registration) so that we can identify your data. None of your e-mail addresses will be shared with any third parties or used for anything other than the aforementioned study.
> 3. We will kindly ask you to send us at least 5 (five) stories of your choice, lasting at least 5 (five) minutes each, that you will have recorded using the application within 5 (five) days. The stories may contain activity logs, which are limited solely to the recording process — we do not get access to your calls, sms, etc. We will however get access to the routes and multimedia comprising the stories. This data will not be shared with any third parties or used for anything other than the aforementioned study.

4. For the duration of the study, you will not be able to share the stories you record, except for sending them to us. The sharing function will be unlocked in the future versions of RouteStory once the study is over.

5. Upon receiving your stories, we will also send you an anonymous questionnaire, including demographics (age, country, etc) and questions regarding your user experience. Filling it in should take no more than 10 minutes. Your personal data will not be shared with any third parties or used for anything other than the aforementioned study.

6. You are more than welcome to report any issues or ask questions at any point.

If you agree to these terms, please submit your e-mail below. Happy storytelling!

Bibliography

[1] Preetha Appan, Hari Sundaram, and David Birchfield. Communicating everyday experiences. In *Proceedings of the 1st ACM workshop on Story representation, mechanism and context*, SRMC '04, pages 17–24, New York, NY, USA, 2004. ACM.

[2] Nicola J. Bidwell, Thomas Reitmaier, Gary Marsden, and Susan Hansen. Designing with mobile digital storytelling in rural africa. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1593–1602, New York, NY, USA, 2010. ACM.

[3] M. Blythe, J. Reid, P. Wright, and E. Geelhoed. Interdisciplinary criticism: Analysing the experience of riot! a location-sensitive digital narrative. *Behaviour & Information Technology*, 25(2):127–139, 2006.

[4] John Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189:194, 1996.

[5] Kevin M. Brooks. Do story agents use rocking chairs? the theory and implementation of one model for computational narrative. In *Proceedings of the fourth ACM international conference on Multimedia*, MULTIMEDIA '96, pages 317–328, New York, NY, USA, 1996. ACM.

[6] Eugene Burmako. Scala macros: let our powers combine!: on how rich syntax and static types work with metaprogramming. In *Proceedings of the 4th Workshop on Scala*, page 3. ACM, 2013.

[7] Vannevar Bush. As we may think. 1945.

[8] Daragh Byrne. *Digital Life Stories: Semi-Automatic (Auto)Biographies within Lifelog Collections*. PhD thesis, Dublin City University School of Computing, 2011.

[9] Daragh Byrne and Gareth J. Jones. Exploring narrative presentation for large multimodal lifelog collections through card sorting. In *Proceedings of the 2nd Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling*, ICIDS '09, pages 92–97, Berlin, Heidelberg, 2009. Springer-Verlag.

[10] Daragh Byrne and Gareth J.F. Jones. Towards computational autobiographical narratives through human digital memories. In *Proceedings of the 2nd ACM international workshop on Story representation, mechanism and context*, SRMC '08, pages 9–12, New York, NY, USA, 2008. ACM.

[11] Antony Courtney and Conal Elliott. Genuinely functional user interfaces. In *Haskell Workshop*, pages 41–69, 2001.

[12] Evan Czaplicki and Stephen Chong. Asynchronous functional reactive programming for guis. In *Proceedings of the 34th ACM SIGPLAN conference on Programming language design and implementation*, pages 411–422. ACM, 2013.

[13] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

[14] Jerry Alan Fails, Allison Druin, and Mona Leigh Guha. Mobile collaboration: Collaboratively reading and creating children's stories on mobile devices. In *Proceedings of the 9th International Conference on Interaction Design and Children*, IDC '10, pages 20–29, New York, NY, USA, 2010. ACM.

[15] Krzysztof Z Gajos, Daniel S Weld, and Jacob O Wobbrock. Automatically generating personalized user interfaces with supple. *Artificial Intelligence*, 174(12):910–950, 2010.

[16] Jim Gemmell, Aleks Aris, and Roger Lueder. Telling stories with mylifebits. In *ICME*, pages 1536–1539. IEEE, 2005.

[17] Colin S Gordon, Werner Dietl, Michael D Ernst, and Dan Grossman. Java ui: effects for controlling ui object access. In *ECOOP 2013–Object-Oriented Programming*, pages 179–204. Springer, 2013.

[18] Anton Gustafsson, John Bichard, Liselott Brunnberg, Oskar Juhlin, and Marco Combetto. Believable environments: generating interactive storytelling in vast location-based pervasive games. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, ACE '06, New York, NY, USA, 2006. ACM.

[19] Marc Hassenzahl and Andrew Monk. The inference of perceived usability from beauty. *Human–Computer Interaction*, 25(3):235–260, 2010.

[20] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. Sensecam: a retrospective

memory aid. In *Proceedings of the 8th international conference on Ubiquitous Computing*, UbiComp'06, pages 177–193, Berlin, Heidelberg, 2006. Springer-Verlag.

[21] Wesley M Johnston, JR Hanna, and Richard J Millar. Advances in dataflow programming languages. *ACM Computing Surveys (CSUR)*, 36(1):1–34, 2004.

[22] Aisling Kelliher and Glorianna Davenport. *Everyday storytelling: supporting the mediated expression of online personal testimony*. Springer, 2007.

[23] Gunnar Liestøl. Aesthetic and rhetorical aspects of linking video in hypermedia. In *Proceedings of the 1994 ACM European Conference on Hypermedia Technology*, ECHT '94, pages 217–223, New York, NY, USA, 1994. ACM.

[24] Siân E. Lindley, Dave Randall, Wes Sharrock, Maxine Glancy, Nicola Smyth, and Richard Harper. Narrative, memory and practice: tensions and choices in the use of a digital artefact. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, BCS-HCI '09, pages 1–9, Swinton, UK, UK, 2009. British Computer Society.

[25] Kevin Lynch. *The image of the city*, volume 11. the MIT Press, 1960.

[26] Joshua McVeigh-Schultz, Jen Stein, Jeff Watson, and Scott Fisher. Extending the lifelog to non-human subjects: Ambient storytelling for human-object relationships. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 1205–1208, New York, NY, USA, 2012. ACM.

[27] Alex Mitchell and Teong Leong Chuah. Telling stories on the go: Lessons from a mobile thematic storytelling system. In *Interactive Storytelling*, pages 83–94. Springer, 2013.

[28] Frank Nack, Abdallah El Ali, Philo Van Kemenade, Jan Overgoor, and Bastiaan Van der Weij. A story to go, please. In *Proceedings of the Third Joint Conference on Interactive Digital Storytelling*, ICIDS'10, pages 74–85, Berlin, Heidelberg, 2010. Springer-Verlag.

[29] Henrik Nilsson, Antony Courtney, and John Peterson. Functional reactive programming, continued. In *Proceedings of the 2002 ACM SIGPLAN workshop on Haskell*, pages 51–64. ACM, 2002.

[30] Pere Obrador, Rodrigo de Oliveira, and Nuria Oliver. Supporting personal photo storytelling for social albums. In *Proceedings of the international conference on Multimedia*, pages 561–570. ACM, 2010.

[31] Martin Odersky and Ingo Maier. Deprecating the observer pattern with scala.react. Technical report, 2012.

[32] K. O'Hara. Narcissus to a Man: Lifelogging, Technology and the Normativity of Truth. In Emma Berry, Daragh Byrne, Aiden R. Doherty, Cathal Gurrin, and Alan F. Smeaton, editors, *Second Annual SenseCam Symposium*, Dublin, 2010.

[33] Bruno CdS Oliveira, Adriaan Moors, and Martin Odersky. Type classes as objects and implicits. In *ACM Sigplan Notices*, volume 45, pages 341–360. ACM, 2010.

[34] Heather S. Packer, Ash Smith, and Paul Lewis. Memorybook: Generating narratives from lifelogs. In *Narrative and Hypertext Workshop 2012*, June 2012.

[35] John Peterson, Antony Courtney, and Bart Robinson. Can gui programming be liberated from the io monad. In *Proceedings of the 2004 ACM SIGPLAN workshop on Haskell*, pages 22–22. Citeseer, 2005.

[36] Gracjan Polak and Janusz Jarosz. Automatic graphical user interface form generation using template haskell. In *Seventh Symposium on Trends in Functional Programming, University of Nottingham, UK*, 2006.

[37] Jason Procyk and Carman Neustaedter. Gems: The design and evaluation of a location-based storytelling game. In *Proceedings of the CSCW Conference on Computer Supported Cooperative Work and Social Computing (CSCW)*. ACM, 2014.

[38] Vladimir Propp. *Morphology of the Folktale*, volume 9. University of Texas Press, 1973.

[39] Meurig Sage. Frantk-a declarative gui language for haskell. *ACM SIGPLAN Notices*, 35(9):106–117, 2000.

[40] Abigail J. Sellen and Steve Whittaker. Beyond total capture: a constructive critique of lifelogging. *Commun. ACM*, 53(5):70–77, May 2010.

[41] Gideon Sireling. Graphical user interfaces in haskell. 2011.

[42] Koh Sueda, Henry Been-Lirn Duh, and Jun Rekimot. Social life logging: can we describe our own personal experience by using collective intelligence? In *Proceedings of the 10th asia pacific conference on Computer human interaction*, APCHI '12, pages 41–50, New York, NY, USA, 2012. ACM.

[43] Philip Wadler. Monads for functional programming. In *Advanced Functional Programming*, pages 24–52. Springer, 1995.