

Context-aware Readings with Inbooki

Daniele Grassi¹, Giacomo Barigazzi^{1,2}, Anas Bouhtouch¹, Giacomo Cabri^{2,*}

¹DM Digital, Modena, Italy

²Università di Modena e Reggio Emilia, Modena, Italy

Abstract

E-books are more and more spreading among readers, who take advantage from the capability of reading them on different devices and of storing a whole library of books in a light and portable device. What is still missing is a technology that enables not only the adaptation to the exploited device, but also to the context in which the books are read. The experience of the users can be more immersive if the flow of the book content can be modified by the reading device depending on different aspects related to the moment and place of reading.

To overcome this limit, we have developed Inbooki, a system for writing and reading e-books that can take into consideration the context in which they are read and can adapt to it, making the experience of the readers more immersive in the story told by the book. The proposed model of the context take into consideration the information from the device sensors, the user profile, and the user choices.

Received on 24 July 2015 ; accepted on 27 September 2015; published on 05 November 2015

Keywords: Context-dependency, E-book

Copyright © 2015 G. Cabri *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi:10.4108/eai.5-11-2015.150602

1. Introduction

An *E-book* is “an electronic version of a printed book which can be read on a computer or a specifically designed handheld device” [4]. E-books can be seen from two points of view: from the one hand, they can be considered as an evolution of classic paper books; from the other hand, they represent a revolution in the way contents are delivered and read [6].

The e-book are spreading, they are likely to be more and more our companions and the trend seems very positive [5, 12]. In fact, today’s e-books have several *advantages* over conventional books; for example: they can be read using different devices and technologies, letting people choose the preferred one; their visualization can be adapted to the devices’ features or to readers’ necessities; a single device can store a great number of books, relieving people from the need of carrying heavy books (or a heavy set of books).

Of course, they can present also some drawbacks or minor problems. Some studies point out that they cannot completely replace conventional books [7] in a very near future. Moreover, some challenges must be

tackled: the studies carried out in [9] and in [13] point out some problems with the technological aspects of e-books, in particular related to the setup process, the processes of finding and downloading the e-books, the hardware, and the reading process itself.

Besides these minor problems, we point out an important aspect; current e-books mainly consist of an electronic version of their paper edition, and this leads to a main drawback: they do not take advantage of the possibilities offered by today’s devices, which can be summarized as follows:

- the chance of gathering information received from *sensors* available in the reader devices (light, temperature, orientation, position, etc.);
- the capability of retrieving information from external *services* (weather, near places, news of the day, etc.);
- the access to the *user’s profile and history* (age, gender, previous readings, etc.).

As a consequence, today’s e-books cannot adapt their content to the context in which they are read, or to the specific user who is reading them.

*Corresponding author. Email: giacomo.cabri@unimore.it

Starting from the above considerations, we designed and built Inbooki, a system to develop and read e-books that take into account the *context* of the readers and adapt the content and the flow to it. So, they can be considered *context-aware* e-books. The contribution of this paper is the detailed description of Inbooki. Some similar approaches exist, but, as we explain in the next section, they are limited because either they are bound to a specific field or they exploit little context information; instead, our approach aims at being more general and at considering many sources of information. Inbooki enables writers to structure their book taking into consideration different aspects of the context where readers will actually read the e-book. The resulting e-books are called *immersive*-books (shortly *in-books*), meaning that the readers are “immersed” in the context where they read the e-books.

The paper is organized as follows. We first provide an overview of the related work in the field (Section 2). Then, we present the model of the context, the structure of an in-book and other details about in-books (Section 3). In Section 4 we explain the architecture of Inbooki along with a presentation of the different components. We will present an example of use of Inbooki (Section 5) and then we discuss about the possible applications of the proposed platform (Section 6). Finally, we discuss results and propose future directions in the conclusive Section 7.

2. Related Work

According to [2], existing systems related to our work can be classified as:

- Location-aware systems, which exploit only a part of the context;
- Context-aware systems, which fully exploit all the context information;
- Context-aware frameworks, which provide an abstract framework for specific context-aware applications.

As said, there are some systems that exploit information about the location (location-awareness) to display customized information to users [3, 8, 11]. Examples of location-aware systems are e-guides for tourists, as the Cyberguide project [1], which aims at building a mobile context-aware tour guide. The implemented system considered several context parameters to adapt the tour to the reader in a form similar to an e-book. However, this project is mostly focused on geographic information related to the tour, and it cannot be exploited for a general-purpose context-dependent book content.

A few examples of context-aware system are reported in the literature. For instance, Muñoz et al. [10] propose

a context-aware system for hospitals that take into consideration several context parameters (not only the user’s location) such as patient turnaround time, the location of other users, devices or artifacts, and the user’s role. This is an interesting approach but strictly oriented to a specific field.

To summarize, the approaches we can find in the literature suffer of being very bound to a specific field or of exploiting little context information. Our aim is to propose a more general approach that considers a wider range of information available.

3. The Immersive Books

3.1. Context Model

The context of the reading is modeled around three sets of parameters: *environment* parameters, *profile* parameters, *reading* parameters.

Environment parameters are the set of external conditions surrounding the user while reading an in-book. Currently, the in-book structure supports GPS localization, weather, air temperature, season and time of the day. Weather and air temperature are determined using third-party web services and GPS location. *Profile* parameters contain the essential data about the user’s profile: gender and age. The user sets these parameters during the registration process, and they can be modified in the reader apps. *Reading* parameters contain the “history” of the user’s reading process for a given in-book: chapters read, reading conditions’ status while reading each chapter, progress.

All these context parameters are used by the Inbooki mobile app¹ to determine which content to show the reader, as explained in the following.

3.2. Structure of an In-book

An *in-book* is an e-book composed of one or more graphs of chapters. Figure 1 reports examples of an in-book graph. Each chapter can have one or more parents and one or more children, and it is composed of an ordered series of blocks. Each block has a specific type (“title”, “text”, “image”) and it can count on one or more variations, depending on the reading conditions. Variations can be grouped in 2 sets: *narrative* and *descriptive*.

Since an in-book is a graph, each chapter can have more than one child: narrative variations occur when the user completes the reading of a chapter and the Inbooki mobile app determines (using the user’s reading conditions) which child chapter to show next among the possibilities. If the reading conditions meet the requirements of more than one chapter, a choice

¹This is the app to read in-books, see 4.4

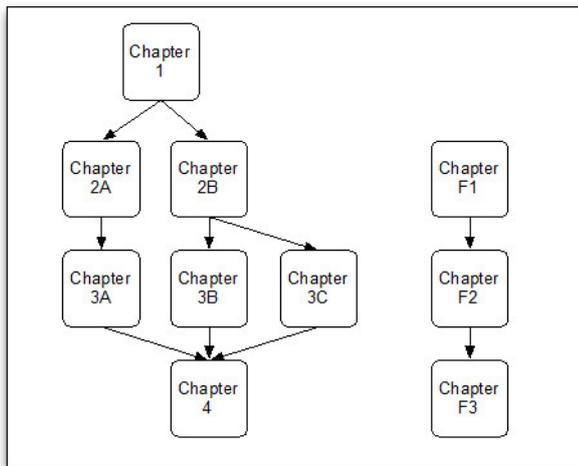


Figure 1. Example of in-book graphs

is presented to the user (a *fork*). If there is only one child chapter and its display conditions are not met, the user is forced to wait (or to take action) until they are met (e.g. the user can be required to go to a specific location, or he can be forced to wait until a specific time of the day comes). Descriptive variations occur inside a chapter's block. Each block can have one or more variations, each one of them having different display conditions. When the reader app has to render a chapter's text, it parses its whole content and determines which descriptive variations have their display conditions satisfied by the current user's reading conditions; if, for a given block, no variation has its display conditions met by the reading conditions, the whole block is not rendered. Figure 2 reports an example of structure of a chapter, writers can define their own structures.

3.3. Free Chapters

As described above, an in-book can be composed of more than one graph of chapters. This means that there can be chapters outside the main graph (being neither children nor parents of chapters belonging to the main graph). Each graph has one or more "root" chapters (not having a parent chapter): root chapters that do not belong to the main graph (called "free chapters") are listed as possible forks when the user finishes reading a chapter and their display conditions are met. For example, a free chapter can be set to become visible as a possible fork when the user is near a specific location at a specific time of the day.

3.4. Next Chapter Selection

When the user starts reading an in-book or finishes reading a previous chapter, the reader apps must determine how to continue the in-book rendering. The

first step is to count how many children chapters have their visibility conditions fully satisfied by the user's reading conditions (environment parameters, profile parameters and reading parameters): if there are one or more results, they will be added to the possible following chapters set; if there are no results, the reader app tries to find the children chapters that have their profile and reading parameters satisfied: if there are one or more results, they are added to the possible following chapters set; if not, a virtual "book-end" chapter is added to the possible following chapters set. At this point the possible following chapters set consists of one or more chapters. A query is made to obtain free chapters that have their reading conditions satisfied, and the results are added to the possible following chapters set. The possible following chapters set is then analyzed: if it consists of two or more chapters, a fork is displayed to the user, who can choose which path to follow. At this point, only one chapter is chosen for rendering. If it is a virtual "book-end" chapter, the book ends. If it is a real chapter, this is further analyzed to determine whether its reading conditions are fully satisfied or not: if one or more parameters are not met yet, a "waiting screen" is shown to the user, who will have to wait in order to continue until the reading conditions are met; if the reading conditions are met, the chapter is rendered.

3.5. Chapter Rendering

Chapter rendering determines the text and the images that are shown to the user once they enter a chapter. The rendering process analyses each chapter's block, verifying that each variation's condition is met in the user's current reading conditions. Blocks that do not have at least one validated variation are not rendered. If a block has two or more variations validated, only the first one is rendered.

3.6. Environment Parameters Detection

The environment parameters can change while the user is reading an in-book. For this reason, the Inbooki mobile apps implement a timed polling cycle that records the parameters' states, which are used by the next chapter selection process to determine which chapters are eligible to be shown to the user when they finish a previous chapter. While time-related parameters are very fast to read, weather-related ones (current weather, current temperature) and GPS location present issues that need a caching system to obtain a smooth user experience. When the application starts, GPS location is usually approximate, and it takes some time for the device to determine the exact position; the Inbooki apps cache the last known location, and they query the device's GPS subsystem during the polling cycle of the environment parameters.

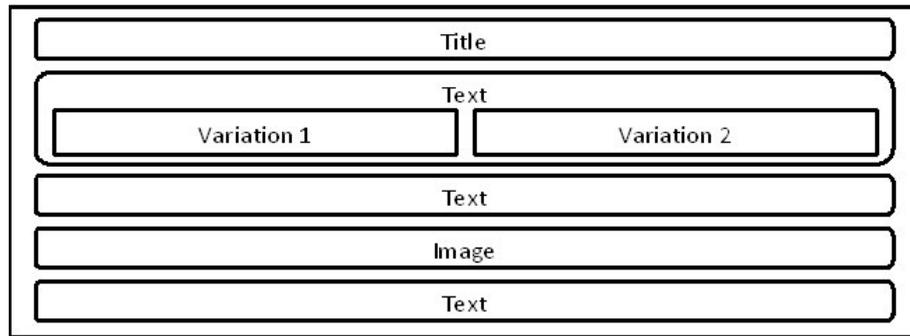


Figure 2. Example of chapter's structure

The same caching system is used for weather-related parameters: as the Internet services used for weather conditions retrieval can be unreachable from time to time (for example when the mobile phone signal or the internet connection are missing), the last weather conditions retrieved are cached for later use, and they are refreshed during the polling cycle of the environment parameters.

4. Inbooki's Architecture

The architecture powering Inbooki is composed of four subsystems:

- the *Inbooki Platform*, which provides API access to user accounts and to the marketplace;
- the *web interface*, which provides users with a website that allows them to register, apply for the Inbooki Author Program and browse the available in-books;
- the *Inbooki Editor*, a web application that allows authors to write in-books in a user-friendly environment;
- the *mobile apps* for iOS and Android devices, which allow the readers to enjoy the in-books using all the supported immersive and interactive features.

4.1. The Inbooki Platform

The platform is the *core* subsystem of Inbooki, providing a substrate which is used by the other subsystems to interact using a private API framework. There are two main API components: *AccountAPI*, which provides access to account utilities such as user registration, login, purchased books listing, account management, etc.; and *MarketAPI*, which provides an interface to the marketplace, including functions like in-book download, in-books information retrieval, marketplace search, etc. API security is ensured by a proprietary OAuth 2.0 implementation, written in PHP.

4.2. The Web Interface

This subsystem comprises a web interface to the marketplace and to account registration and management utilities. The main endpoint for the web interface is the Inbooki website (<http://www.inbooki.com>), which was developed using HTML, PHP and JavaScript using a Bootstrap template.

4.3. The Inbooki Editor

The Inbooki Editor plays an important role in the Inbooki architecture: it provides the authors with a user-friendly web application that allows them to write in-books in an easy-to-use yet complete environment. Using the editor, the authors can add *immersive* and *interactive* features to their in-books using a graphic interface. A built-in compiler gives the authors the possibility of creating deployable in-books that can be read on mobile devices running the Inbooki app for iOS or Android.

Using the editor authors can submit their in-books to the marketplace. The editor is written in PHP, HTML and JavaScript, using additional libraries as jQuery to enhance the user experience.

Snapshots of the web interface will be shown in Section 5.

4.4. Mobile Apps for iOS and Android Devices

An in-book contains several immersive and interactive features that are not supported by current e-books standards like ePub, mobi, etc. This is the reason why conventional e-book readers cannot display an in-book. iOS and Android Inbooki apps fill this gap, providing a reading experience that is extremely similar to the one provided by classic e-book readers (e.g. the user swipes to go to the next page), while introducing immersive and interactive features that allow the readers to fully enjoy the possibilities offered by the in-books.

The Inbooki apps give the reader the possibility of downloading the available in-books and automatically

syncing the reading progress on all their devices (e.g. a given reader can use the same Inbooki account on an iPhone and on an Android device).

The immersive variations to the in-book's text—such as variations determined by weather conditions, time of the day, day of the year, user profile, etc.—are automatically processed during the reading, and each chapter is compiled on the fly using the conditions set by the author and determined by the reader's environment: the story itself, or single paragraphs, can thus change according to these conditions.

The Inbooki apps rely mainly on GPS localization to detect the reader's environment and to define the context: the data received is used to interact with external services to retrieve weather conditions and temperature, as well as to determine the “visibility” of geolocation-conditioned chapters (chapters that can be read only if the reader is within a certain distance from a specific geographical point).

The user's reading “history” is recorded and can be used by the authors while writing the in-book to determine story variations according to the path followed by each reader: for example, the story can change according to the path the reader chose in earlier chapters.

In-books also present interactive features: the author can design *forks* along the story where the reader is actually asked to make a decision. When the reader encounters a fork, the mobile app shows them the list of possible decisions and then calculates the following chapter according to this choice.

5. Example of Use and Impact

In this section we show how the Inbooki system can be exploited to write and to read in-books. We provide also an overview of possible applications.

5.1. Developing an In-book

To develop an in-book, the writer must first fill in the text of the chapters (see Figure 3)² as in a traditional editor.

The chapters have a customizable display order in the editor (see Figure 4) which doesn't necessarily correspond to the narrative order that is seen by the readers. For instance, the chapter “Scegli di confidarti” is a sibling of “Rispondi che non te lo ricordi”: they both follow “Colazione in famiglia” (the reader is shown one of them depending on his context), but they continue in different narrative threads.

This leads to a non-linear chapters graph (see Figure 5); using the editor the writer can keep under control the structure of the in-book.

Furthermore, different conditions can be associated to each chapter. Figure 6 shows a dialog in which the following conditions can be defined:

- time of the day;
- weather conditions;
- age of the reader;
- external temperature;
- season;
- gender of the reader;
- location.

Advanced conditions can be specified as well.

Figure 7 details the capability of setting geolocation conditions for a chapter: the writer can decide where in the world a given chapter can be read or, form the point of view of chapters, which ones will be triggered by a reader's given location.

5.2. Enjoying an In-book on Devices

We are aware that it is impossible to provide readers with the true experience of an in-book, since it is very dynamic, anyway in this section we sketch how the readers can enjoy the in-book using a mobile device (in this case, an Apple iPhone). As it can be seen from Figure 8, the reading of in-books is very similar to the reading of standard e-books.

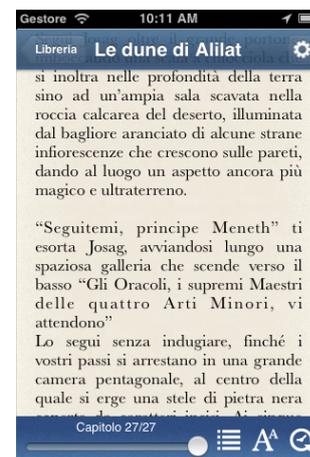


Figure 8. Reading interface (with menu)

One of the specific enhancements offered by in-books is the possibility to set some parameters before starting the reading; Figure 9 reports the reader's choice of gender, age, weather conditions, hour of the day, season, and temperature. The values of all these parameters can be defined in an automatic way: they can be retrieved from sensors, from (web) services, or from the user's profile.

²The screenshots are from the Italian version of the software.

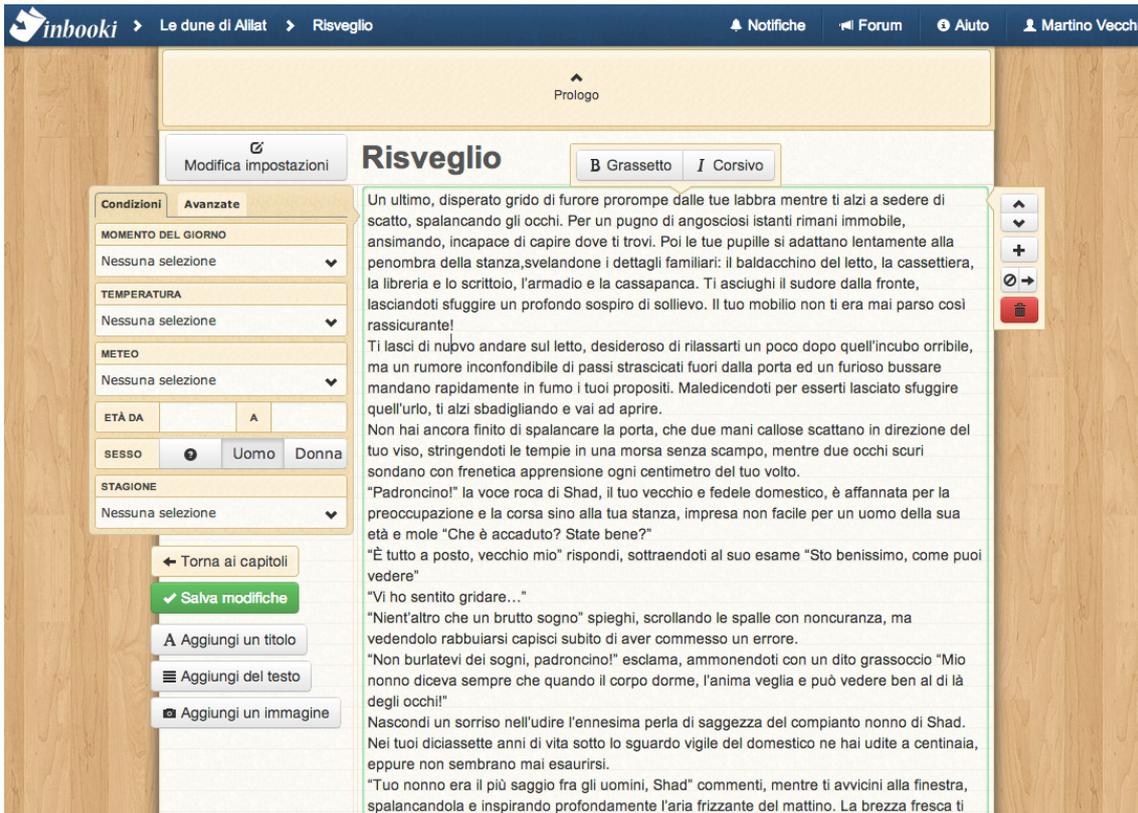


Figure 3. Inbooki chapter text

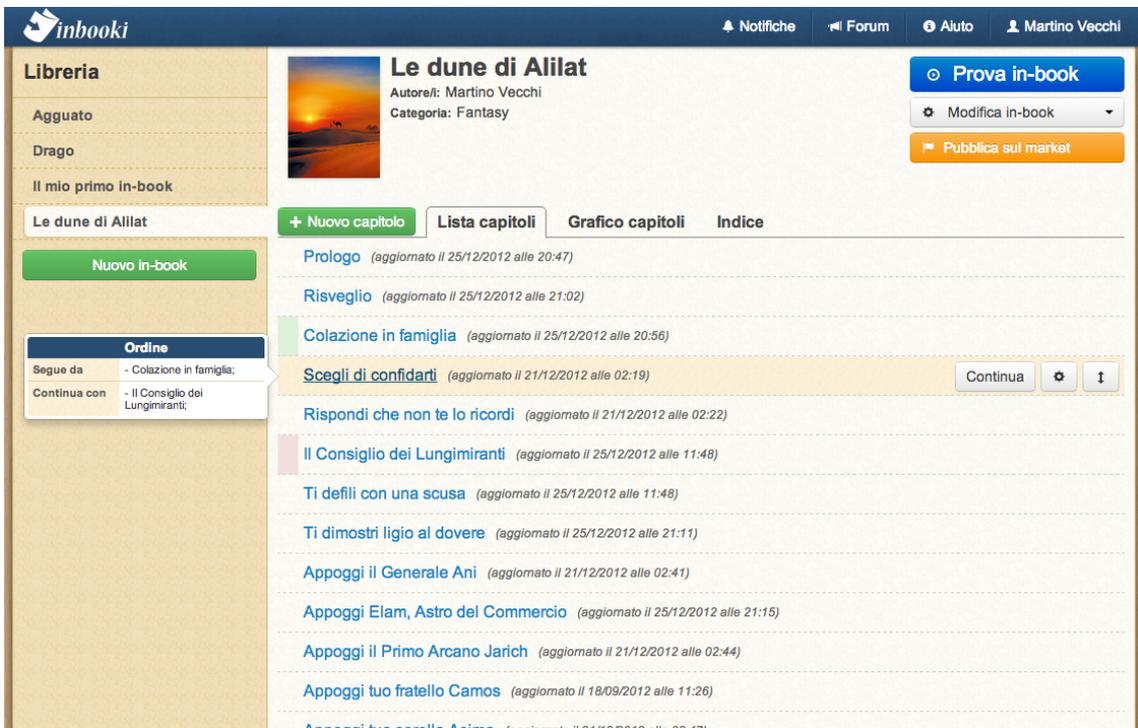


Figure 4. Inbooki chapters

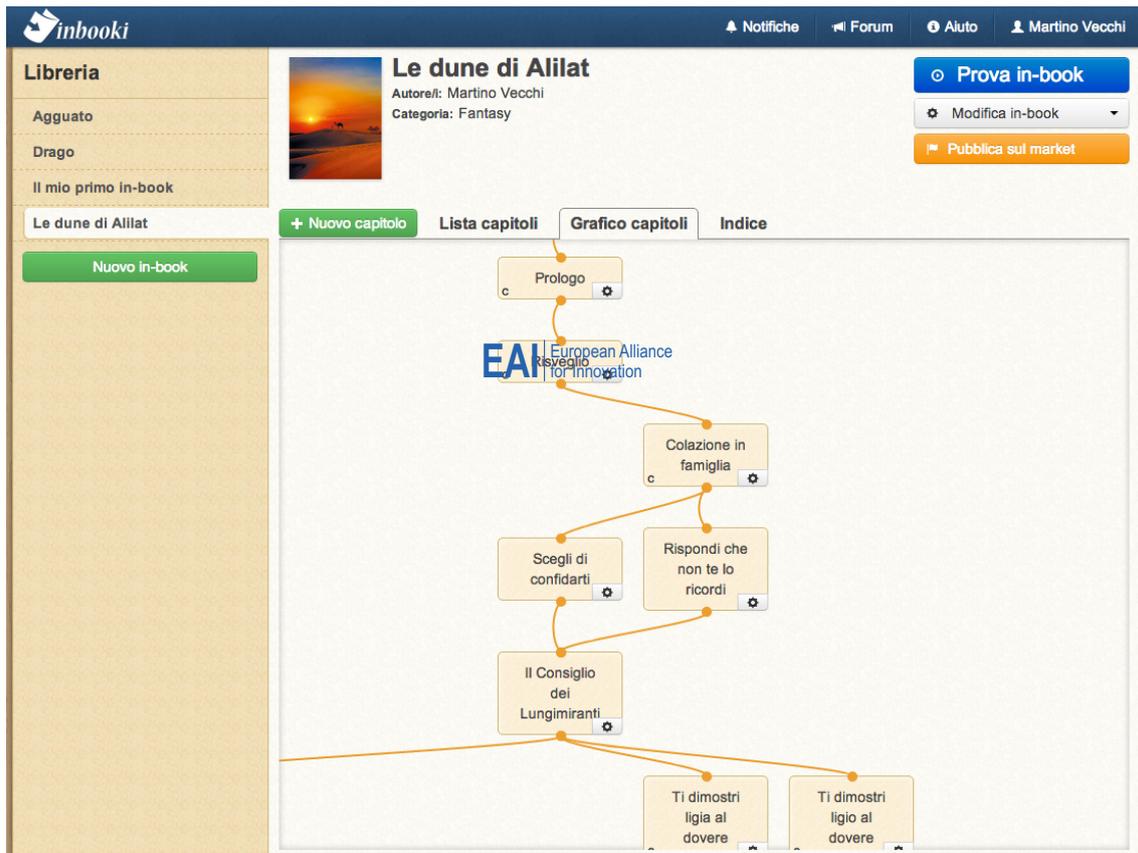


Figure 5. Inbooki graph

Figure 6. Inbooki chapter conditions

In Figure 10 we show a distinctive feature of in-books: facing a fork, the reader can choose which path to follow, which will lead to two different chapters, according to the structure of the in-book as defined by the writer.

6. Possible Applications of Inbooki

From a literary point of view, Inbooki introduces a new kind of e-book, in which stories can unfold in different directions depending on the user's choices and

environment. An in-book could be seen as an enhanced version of the old gamebooks, quite popular in the '80s. Using the Inbooki Editor, the creation of an interactive and immersive e-book does not require any technical skills, and this represents an opportunity for authors to express their creativity without resorting to software developers to create their interactive e-books.

A practical example of use of Inbooki is found in the *guidebooks* field: currently, the creation of an interactive guidebook, able to use the reader's location

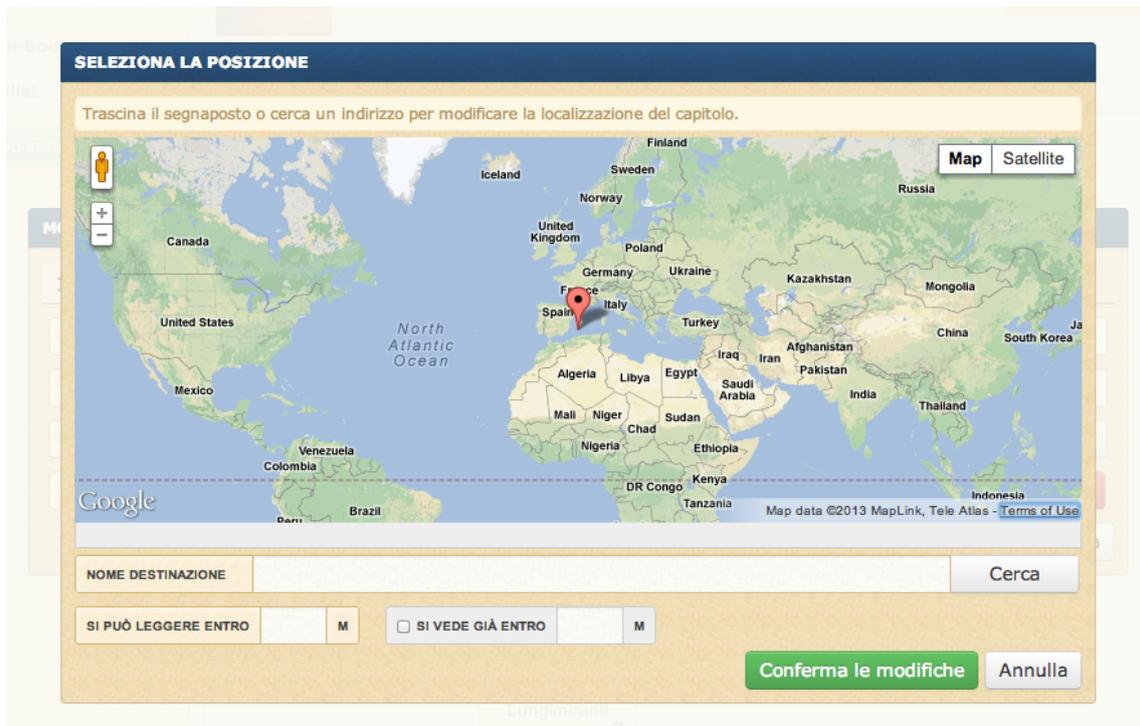


Figure 7. Inbooki localization



Figure 9. User's profile



Figure 10. Reader's choice

to provide enhanced information, is restricted to who is willing and able to contract a software house and develop a dedicated mobile app. This means having to cope with the difficulties of deploying and distributing an app among different platforms (iOS, Android), and to depend on the software development team to provide the users with the updated content. Inbooki allows guidebook authors and editors to create their in-books using the Inbooki Editor and distribute them automatically to iOS and Android mobile devices, without needing any technical knowledge. The updating of the in-book content can be done using the

Editor, and the release of new versions of the in-book is notified automatically to users, who can download it to their devices. In the guidebooks' field, an in-book can provide immersive and interactive e-books that can suggest to the user different tours and paths depending on the environment (e.g. the weather), on their personal interests (using forks) or their profile (gender and age). Using free chapters, specific content can be displayed depending on the location of the user and the environment conditions.

The concept of in-book can have a big impact on the e-book market, creating a niche for interactive

and immersive stories and guidebooks. The supported features of the in-book can be easily further extended, thanks to its open structure which allows different types of blocks to compose a single chapter, and thanks to the flexible format of chapters' conditions. In the future, for example, video support could be introduced, or web-enhanced blocks could be defined to integrate web pages inside chapters.

Outside the literary field, Inbooki can be a useful tool to create simple context-aware applications without the need of any specific technical knowledge. A typical application could be the creation of business scenario tests, where MBA students have to make business decisions based on the information provided in the text. These business scenario tests could be easily developed by teachers using the Inbooki interface, making use of the "fork" features of in-books, or provided directly by authors or publishers.

7. Conclusion and Future Work

Inbooki is a system to develop adaptive e-books that take into consideration the context where they are read and which allow user interaction. These e-books are called "immersive books" or *in-books* because they consider the readers as "immersed" *in* the context in which they read the e-books themselves.

In this paper we have detailed different aspects related to Inbooki, from the model of context we have considered, to the architectural components of the system. Moreover, we have reported about the tools available for writers and readers, and we have proposed some uses of this kind of e-book, even if many others can be possible. The availability of these e-books will make people aware not only of their existence, but also of their usefulness, creating the basis of an interesting market.

Regarding future work, the first aspect on which to focus can be how to effectively advertise the immersive books. From the technical point of view, today's e-readers and in particular smartphones are equipped with a broad range of sensors, whose contextual information can be taken into consideration by the authors to define the in-books' behavior. In a long-term work, interactivity can be enhanced enabling the readers to directly add content to the in-books, and encouraging the author-reader relationship adding specific communication tools to the Inbooki apps. Finally, also non-alphabetic language could be supported to make in-books available worldwide.

Acknowledgement. This work is supported by the project "Algorithms and Models for Building context-dependent Information delivery Tools" (AMBIT) sponsored by Fondazione Cassa di Risparmio di Modena (prot. Nr. 188.14.8C, SIME 2013.0660). Website:

<http://www.agentgroup.unimore.it/ambit/>

References

- [1] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, 3:421–433, 1997.
- [2] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2:263–277, 2007.
- [3] Jenna Burrell and Geri K. Gay. E-graffiti: evaluating real-world use of a context-aware system. *Interacting with Computers*, 14(4):301 – 312, 2002.
- [4] Oxford Dictionary. E-book, 2015.
- [5] R. Duncan. Ebooks and beyond: The challenge for public libraries. *Australasian Public Libraries and Information Services*, 23(2):44, 2010.
- [6] D. Živković. The electronic book: evolution or revolution? *Bilgi ḋinyasi*, 9(1):1–19, 2008.
- [7] Y.Y. Kang, M.J.J. Wang, and R. Lin. Usability evaluation of e-books. *Displays*, 30(2):49–52, 2009.
- [8] Clemens Kerer, Schahram Dustdar, Mehdi Jazayeri, Danilo Gomes, Akos Szego, and Jose A. Burgos Caja. Presence-aware infrastructure using web services and rfid technologies. In Jacques Malenfant and Bjarte M. Østvold, editors, *Object-Oriented Technology. ECOOP 2004 Workshop Reader*, volume 3344 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004.
- [9] P. Lam, S.L. Lam, J. Lam, and C. McNaught. Usability and usefulness of ebooks on ppcs: How students' opinions vary over time. *Australasian Journal of Educational Technology*, 25(1):30–44, 2009.
- [10] M.A. Munoz, M. Rodriguez, J. Favela, A.I. Martinez-Garcia, and V.M. Gonzalez. Context-aware mobile communication in hospitals. *Computer*, 36(9):38 – 46, sept. 2003.
- [11] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00*, pages 32–43, New York, NY, USA, 2000. ACM.
- [12] A. Soules. The shifting landscape of e-books. *New Library World*, 110(1/2):7–21, 2009.
- [13] W.D. Woody, D.B. Daniel, and C.A. Baker. E-books or textbooks: Students prefer textbooks. *Computers & Education*, 55(3):945–948, 2010.