

Designing a Viral Collaborative Tool: Patterns and Guidelines for Virality-Driven Design Master Thesis in Interaction Design and Technologies.

Juan Carlos Wendeus

Department of Applied Information Technology CHALMERS UNIVERSITY OF TECHNOLOGY Gothenburg, Sweden, 2013 Report Number: 2013:130

Designing a Viral Collaborative Tool: Patterns and Guidelines for Virality-Driven Design © JUAN CARLOS WENDEUS, 2013

Report Number: 2013:130

Department of Applied Information Technology Chalmers University of Technology SE-412 96 Gothenburg Sweden Telephone + 46 31-772 1000

© Juan Carlos Wendeus Gothenburg, Sweden 2013

Acknowledgements

This thesis has been an interesting challenge that taught me a lot about research techniques and how to organize masses of information in a valuable way. For that I would like to thank my thesis supervisor **Sus Lundgren**, also for her patience and great tips to improve this project. Additionally I would also like to thank **Joel and Emma Rozada**, because of their help, this project was possible. They introduced the concept of virality and they were incredibly kind and helpful throughout the process. Finally, I would like to thank you my lovely wife **Monica Esteban**, for her patience and support spending many weeks along with me working in this project.

Abstract

In the last decade technology has changed the way people behave, allowing them to share more information quick and easy, creating a phenomenon where a piece of information can be spread massively. This effect is known as a viral effect. Thus, when people talk about viral products or services they refer to designs that display a certain *marketing strategy*, which aims for such a viral propagation among users. However, there is no record of a design perspective to address this issue, i.e. how to *design* products or services that imbue some sort of inherent virality. Consequently, this thesis aims to explain and exemplify how services or products can be designed to be viral through a collection of patterns and design guidelines.

The thesis had two aims: to explore how to design for virality, and to design a viral collaborative tool. Therefore, it provides a collection of patterns, which were collected in an extensive literature study. This collection, together with patterns for collaboration, was used in an iterative design process in order to create several prototypes from low-fidelity to hi-fidelity responding to the second task. Finally, the third task was to perform an evaluation in regard to the first aim: to test guidelines and patterns proposed in the document. Unfortunately, as is mentioned in the discussion, a test was impossible to carry out, since the design was not implemented and some of the outcomes of this research were that virality is quite related with to temporal phenomena and real-life interaction, making it difficult to test a design unless launched in a real environment where the users can decide and spread the word freely without any pressure or vicious information. However, the sponsor company approved of the prototype, and it will possibly be implemented to test its viral approach in the market.

Keywords: Virality, Stickiness, Patterns, Interaction Design, GUI Design

Contents

Acknow	ledgements	3
1. Intr	oduction	1
1.1.	Problem / Project Aim	1
1.1.1	. Tasks	2
1.1.2	Research Question	2
1.1.3	S. Structure of the Report	2
1.1.4	Planning the Process	2
2. Bacl	sground	3
3. The	orv	4
3.1.	Virality	4
3.2.	Stickiness	5
3.3.	Collaboration	7
3.4.	Patterns	8
3.5.	MDA Model	8
The M	DA of viral patterns	9
4 Met	hods	12
4 1	l iterature Search	12
4.2	Pattern-hased design Annroach	12
4.3.	Iterative Design	13
4.4.	Prototyping	13
4.5.	Personas	13
4.6.	Use Cases	14
4.7.	Target Users	14
5. Proc	cess	15
5.1.	Patterns	15
5.1.1	. Viral Patterns	15
5.1.2	2. Collaborative Patterns	16
5.2.	Using and classifying Patterns	17
5.2.1	Using MDA Model	17
5.3.	Design Process	19
5.3.1	. The Application	20
5.3.2	L. User Scenario, Target Users	20
5.3.3	5. Design Iterations	ZZ
6. Rest	1lts	31
6.1.	Stickiness Rules	31
6.2.	Viral Patterns	31
6.3.	Collaborative Patterns	41
6.4.	Patterns within MDA model	41
6.5.	Design Guidelines	42
6.6.	Final Design	45
7. Disc	ussion	57
Lookir	ng for Virality	57
The Pa	ttern's approach	58
The M	ethods	58

A	bout the results	. 58
F	uture considerations	. 59
8.	Conclusions	61
9.	References	62
10.	Appendices	65
Α	ppendix A	. 65
A	ppendix B	. 66

1. Introduction

In recent years the Internet has grown exponentially, introducing new technologies such as cloud computing, social networks and collaborative tools, promoting the development of collaborative work with people in different parts of the globe, sharing information and improving the workflow of companies and people itself. These technologies have become a commodity available for everyone who wants to use it. For that reason companies has a fierce competition; forcing them to act in creative ways to attract people to their services. In the majority of the cases services are design based on certain requirements but then a marketing team creates some strategies to make this service in something that spreads rapidly among customers in order to make it popular and creates a viral phenomenon among the target group.

During the spring of 2012 was carried a thesis at and for *Nordiska Interaktionsbyrån* (NIB) about the virality phenomena; concluded in a set of rules defining what and how certain function imposed in a product can help to spread and enhance a viral behavior. NIB had a vision to go to the next level continuing with the work performed in the previous thesis; researching for design guidelines in collaborative environments and creating a prototype of a cloud based service for collaborative environments that apply the rules of virality proposed by (Wodtke, 2009) and (Rodriguez, 2012). The main idea was to create a mashed-up solution that contains some functions that appears in a Taking-Note, Keynote, GTD¹ and Calendar Applications producing a novel solution for meeting and group activities that could spread quickly among a target group.

1.1. Problem / Project Aim

Usually, when people talk about viral products/services, they refers to products/services that have certain marketing strategy imposed that have as effect, a viral propagation among users. However, there is no record of a design perspective for this issue; where the product is designed in purposed based on viral rules and patterns.

Consequently, this thesis aims to develop an understanding of what causes that services and/or applications become viral. Thus, this knowledge could be used in creating rules and design patterns. This can be reproduced and applied in other products or services within the company.

¹ **GTD:** "The Getting Things Done method rests on the idea that a person needs to move tasks out of the mind by recording

1.1.1. Tasks

- a. Create the first set of theoretical design guidelines for a collaboration tool, taking into account the previous research about virality.
- b. Based on the above research, design a prototype for an online viral collaboration tool in the form of an elaborate GUI framework, however not implemented.
- c. Evaluate the design in relation to the guidelines and theories found in step 1. As well as in relation to usability and functionality.

1.1.2. Research Question

How can we, in design, utilize virality as a means to drive a computer-based collaboration among users?

1.1.3. Structure of the Report

This document is divided in four main sections. First Theory; this section will describe the different theories used as foundation for this project. Following that the Methods used to carry on the project. Then continuing with the Process done during the thesis and finally, the results and discussion where are described in details the outcomes and the implications of this research.

1.1.4. Planning the Process

Before the actual work began, the need for plan and schedule was necessary. The plan and schedule was done in collaboration with my supervisors in NIB and Chalmers. Since it was very complicated have time-knowledge consumption in advance. We plan a tempting schedule (See <u>Appendix A</u>) where the tasks are divided in:

- a. **Preparation (4 weeks):** In this part of the project were carried will start with an exploration of the literature search and starts with the theory around the thesis project.
- b. **Design Research (9 weeks):** In this stage most of the literature search is done and is required to the creation and description of the Design Patterns.
- **c.** Design and Concept Proposal (4 weeks): In this stage the project is required to start working with the design of the viral collaborative application based on the outcomes of the Design research.
- d. **Evaluation (1 week):** In this phase the design should be evaluated in order to analyze the Design Research done previously in the project.
- e. **Refinement and Conclusion (2 Weeks):** These two weeks should be used to refinements of the Thesis report and writing conclusions and discussion around this project.

2. Background

During the last decade many changes in technology have occurred (Whittaker, 2009), creating a fascinating scenario; where people are connected through devices like mobile phones or tablets constantly sending and consuming information from each other. This has generated a phenomenon where some pieces of information are scattered rapidly among people creating a "contagious" effect. In other cases this happens also with certain services or products like "Instagram" or "Dropbox" that have fast adoption rates making them contagious too.

This contagious effect is well known in the Marketing jargon as a viral effect (Godin, 2000). This viral effect happens when a product or service has a fast and enormous adoption rate among its potential users. Usually those adoptions rates happened during its launching into the market. But in other cases it could happen sometime after the product is launched.

Companies contract advertisement agencies that help to create "Viral" strategies taking a product/service already designed and defining some kind of strategy around it, to make it viral. But what if instead of that, these viral phenomena were used as a design tool? Designers could have a powerful tool that helps them to create valuable and thoughtful products that use virality as a method to create them; making these products before be launched into the market "Viral Enabled" instead of the old method done by advertisement agencies.

During the research carried out for this project, it was found that there is not such a thing like design guidelines for virality or designing for virality. Most of the content and literature available about this topic is based on a business or marketing point of view. Thus, due to this, there is not that much work related to this topic. However, many researchers have thought in similar scenarios that touch the surface of this concept like (Wodtke, 2009), (Fogg, 2003), (Anderson, 2012) that have work about topics like what enable people to use some service and how certain element can motivate people to share and spread the word around an specific group. So, taking them and other authors in to account in this document, this thesis project was based.

3. Theory

When designing virality from an Interaction Design point of view there were a few concepts that came to be the basis of the research approach taken in this thesis. These concepts are divided in theoretical and structural. The theoretical concepts are Stickiness, Virality and Collaboration and for the structural those are Patterns and MDA model. Thereby, the theoretical group offers the concepts necessary to support this thesis and the structural provides a method to arrange the theoretical concepts in a hierarchical way making easy in the future for other designers make use of them. Additionally, this theory brings the Virality concept that is the backbone of this project. Below, these concepts will be described in detail.

3.1. Virality

Virality refers to the phenomenon where services or products present an exponential growth within a community or a specific group (Wodtke, 2009) other authors define virality as an idea that spreads and it is adopted by many.

Nowadays consumers show more and more resistance to traditional forms of advertising, this combined with an exponential growth of devices connected to Internet have created the conditions for users to avoid traditional advertising by using these other devices that have a better offer than the traditional. For that reason companies are trying to look for alternative ways to reach people in order to sell their products. They have tried approachable methods in where they can be constantly connected allowing the same users work as sellers, spreading the word about the qualities of those products and making them viral (Leskovec, et al., 2007).

Thus, When Virality is present is also related with the phenomenon of Word-of-Mouth (WOM) in which a person passes the word about a product or service to other creating an exponential chain where people find out the qualities of it through others instead of the traditional marketing channels (Goldenberg, et al., 2001).

This phenomenon creates a cycle where the more people spread the concept, the more people see it; thus, the more people see it, the more people spread it (Godin, 2000). But, in the Interaction Design area it has not been that many approaches except for just a few authors that have work around the "viral" concept in different design levels; for example, Christina Wodtke (2009) has performed her investigation around the notion of patterns that are usually present in application that spreads virally, like Instagram. Those patterns are usually related with the simplicity of the design; how the designs create a flow that allows the user share and spread the word without frictions or how the design should be focus in a specific target group. Additionally she

has made use of known concepts like *At Hand* where the most important elements of an interface should be present close to the user working area in order achieve certain action, this is also named in the book "About Face" (Cooper, et al., 2007).

Other authors, like Goldenberg et al (2001) and BJ Fogg (2003) have explored the phenomenon where certain action can have repercussions creating bigger events. Making a simile of this, it is similar to the effect done by a rock when smashes against water body creating a small ripple to then increase their effect becoming a bigger wave. In other words a when a small piece of information is shared to a group this one can be copy and reproduced that many times that could cover a huge mass having a significant effect in the group.

Moreover, the discussion is also centered in how the service or product not only allowed promoting certain piece of information to make it viral. Actually at the same time when the service makes viral stuff the service by itself is viral. Said in other way, virality is applied not only to information is applied to the service that is enabling the viral information. For example, when people use Instagram and one of the photos becomes really popular; the photo is viral. But, Instagram is viral too because people need to use it in order to look the viral photo. This is one of the statements discussed by Seth Godin (2000) and Christina Wodtke (2009) where they formulate the idea that services contain tools that allow the self-promotion of the service.

The discussion related with virality sometimes is often related with information and services creating stuff to be shared through several means. But, what about human behavior this is also a key component of the viral process. Actually virality as is expressed in this document it would be nothing without human action. So, Kollock and Smith (1999) talk about the concepts of reputation and reciprocity as viral drivers too. So virality is not only actions enable in services are also relations with other people where unstated responsibilities to repay and the social hierarchies presents certain value to create connections and relations quickly. Furthermore, concepts like *Showing-off* and *Complaining* about something are also human behaviors that let establish relations among people. For instance, when famous people share information about how good they live, other become in followers transforming this cult of the desire for someone else fortune a good way to create viral movements. On the other hand, Complaining seems the other extreme of the human behavior where people have a common enemy that is used to attract the attention making this action *seductively* viral too (Anderson, 2012).

3.2. Stickiness

Stickiness refers to a service, product or content designed to hold the user's attention, spending longer periods of time using this service or product. This concept is explored and discussed by Heath and Heath (2007) in their book "Made to Stick" and Malcolm Gladwell (2002) in his book "The Tipping Point"; there the authors look for different examples where the stickiness is present in order to find an explanation to this common phenomenon of spread ideas.

Thus, authors explore the social phenomena of how thing could spread from small group to big groups in a contagious, a viral way. They look for factors that make products or services sticky bringing a set of rules to follow in general when it is designed a product. They have proposed a model where the *stickiness* of an idea, product or concept is based on six basic principles: *simplicity, unexpectedness, concreteness, credibility, emotions* and *stories*.

- **Simplicity:** A simple idea is easier to spread, explain and memorize among the people. Thus, if the concept, idea or service were easy and simple to understand, "to digest" and to use. It is quite likely that people spread and follow a product with those characteristics easily without doubts.
- Unexpectedness: Surprising the user, coming with new concepts creates an emotion of alertness and focus. As the Heath & Heath have stated in their book (2007) quote: "We need to violate people's expectations..." This means the need to break into people lives with simple concepts but unexpected generating interest and curiosity.
- **Concreteness:** This principle refers to the notion of create specific messages that are not ambiguous for the users. In other words a concrete idea that explains easily what the concept is and what do and do not.
- **Credibility:** Sticky concepts, products or services have to bring with their own credentials. Thus, if the products have to convey credibility is necessary let people test by themselves what is saying about the product.
- **Emotional:** People are wired to feel things for other people, not for abstractions or vague concepts. Thus, the service/product have to exploit in some sort this concept letting people connect with other and create emotional relations.
- **Stories:** Stories are always a good way to conduct and act as a mental simulator for situations. In that way when people that hear stories from others are preparing themselves for a new experience, a new action or a new product. Letting them use this principle as an introduction to something.

These principles are very important when designed for virality is. They are essential concepts that used correctly can help to trigger and motivate certain desired actions in the service or product. Then, it is important to always consider them when is designed for virality. In that way, they should be considered as meta-patterns that are permanently present in the design concept ensuring the stickiness values of the service or product.

3.3. Collaboration

Collaboration is a process that requires two or more individuals to generate a useful action through communication. Collaboration is based on a common goal, with certain objectives and a defined outcome. Additionally, collaboration is an elaborated process of negotiation to fulfill and accomplish the main goal (Dunham, et al., 1986) Additionally, computer based collaboration has a structure based on different steps of the process where people perform certain actions or some behaviors are expected to fulfill the aim of the collaborative process. These steps are discussed by de Moor (2006) where he establishes five groups of patterns to identify common elements in the collaborative environment. These groups of patterns are divided in Goal patterns, Communication patterns, Information patterns, Task Patterns and Meta-Patterns. These groups of patterns in turn are sub-divided in some others patterns with specific description of general issues commonly presented in a collaborative environment. For example, in the Goal group Annotate is discussed by Iacob (2011) as a manner to call back thought or ideas in an understandable way. But, this pattern is connected with a Structured View letting the user have an understanding of the document shared, helping to fulfill the goal of the collaboration. Moreover, Patterns like Collective Intelligence, Peering and Folksonomy (Stewart, 2008) are also enablers of the goal proposed in the collaboration.

Then, collaboration is not possible if there is not *communication* between the coworkers. Thus, some authors like Stewart propose in his book Wikipatterns (2008) patterns such as *Socialize, Recognition, Kick-Off* and *Communication* as common issues that have to be treated during a collaborative situation.

Information basically is the knowledge transferred in some particular fact related with the collaboration. Hence, the collaborative process requires to be solved certain structure and flow. These *information* structures can be defined creating certain *Hierarchy* (de Moor, 2006) in the document. Also, having a *Preview* (Schummer, 2004) the information could be visualized by other in a neat and hierarchical way, simplifying the collaborative process. However, the information sometimes requires certain clarification in complicated situations, for that reason *Helping* is needed in order to help users in difficult situation where the information is to complex or the collaborator is new in collaborative team (Schummer, 2004).

In contrast, the information generated during the collaborative process does not have any sense if it is not use to accomplish certain *tasks*. Mader Stewart (2008)has proposed that *Scheduling, linked Information* and *Team-Up* are common patterns that allow the user create tasks looking for the main purpose established in the collaboration practice. In addition, Schummer (2004) mention *Assigning* as the way to leave the responsibility to others to fulfill the aim of the final collaborative purpose.

Finally, all the collaborative patterns have common values that are called met-patterns because they are present in some way or other in them. These meta-patterns are

Versioning (Iacob, 2011), *Consensus* (Schummer, 2004) and *Sessions* (Schummer, 2004) and they discuss about the mechanisms that users have to control the different copies of the collaborative document and how people define an agreement in a topic discussed during the collaborative course.

3.4. Patterns

Patterns in this thesis have become the corner stone of the project, offering a useful tool to organize and hierarchize the information obtained in the literature search. Patterns as a design solution have seen its origins as an answer for typical architectural problems by Alexander et al (1977) in the book "A Pattern Language". Then, this approach was broadly used in computer sciences by Gamma et al (1994) in the book "Design Patterns: Elements of Reusable Object- Oriented Software". Currently other authors have make use of this method commonly in an Interaction design approach like (Borchers, 2001), (Crumlish & Malone, 2009), (Tidwell, 2005) and others in Game Design like (Bjork & Holopainen, 2005).

In this thesis the patterns where describe based in a structured inspired by the book Game play design Pattern of Björk & Holopainen (2005), where they structure the patterns with a common description of the issue presented and then is exemplified by known cases to finally offers a method of how to use the pattern. Additionally the patterns presented in this thesis are arranged based in their position over the MDA model (Hunicke, et al., 2004), arranging the aesthetical patterns in the top, continuing with the Dynamical patterns and finishing with the mechanical patterns.

But, why is it used patterns? As Crumlish and Malone have described in their book Designing Social Media Interfaces (2009); A Pattern is a common successful design solution for a known problem in a context. Additionally, they can be combined with other patterns to create reach experiences among the users. So, is possible to say that patterns do not work as isolated pieces. Instead, they works similar like Lego pieces that support one with the other creating complex structures to solve complex design problems. And because of the aim of this thesis this seems a valuable solution to be used in other to fulfill the purposes of the stakeholders. Then, patterns are presented usually as simple descriptions that could be used as a guidance to solve specific problem in particular contexts.

3.5. MDA Model

MDA is the acronym for Mechanics, Dynamics and Aesthetics. Hunicke et al (2004) proposed this model; as a formal approach to game design and game research, creating an interesting technique to understand games not only from the developer view but also from the designer and user view. Although, this approach comes from the game world, it works as a great connector for the pattern collection presented in this document.

The MDA model refers to a framework that supports an iterative approach to design, which allows thinking about particular design goals and anticipating how changes will impact into the resulting design. Additionally, it helps to provide a holistic view of cause and effect into the design.

This acronym (*Mechanics, Dynamics and Aesthetics*) have three specific parts that explains clearly how is affected the design. Thus, Mechanics refers as the particular component of the design that operates the rules. For instance, from a software point of view mechanics refers to what is encoded in the application. Dynamics refers to the behaviors of the users. Again in the software example, means how people use the application, this include all the uses intended or unintended. And finally, Aesthetics that describes the desirable emotional responses evoked in the user. Usually this could be understood as the vision behind the design.

The MDA of viral patterns

A good example of an application that has the MDA model present and uses some of the Viral Patterns is a web application called MINUTES. IO^2 . This application is designed to take notes and share them right away. If it is taken a look based on the viral patterns inscribe in the MDA model, is possible to argue that *simplified (A)* and *frictionless (A)* are directly intended as Aesthetical patterns that are aspiring to enable *team-up (D)* and *collective Intelligence (D)* by the use of mechanical patterns like *stackability (M), template content (D)* and *Invitation (D)*. In this sense the MDA model let the designer to have an understanding of the flow intended to achieve and trigger certain behavior in a service or product.

Now, looking in detail is possible to appreciate the green button in the home page which let the user work immediately with the application making it *Frictionless* (A) in its use and *Simplified* (A) for the user since he/she can try it without any compromise (See Screen 001).



 $Screen \ 001-Frictionless \ button.$

² Minutes IO: <u>https://www.minutes.io/</u>



Screen 002 - Main Screen

Then when the application begins the user the whole interface is so simple that is meant to be *Irresistible (A)* used. The whole system let the user contact and invites other people based on other services creating a *Stackability (M)* of connections that help to spread the word of this service among a group (See Screen 002). Additionally, the system memorizes the user even if this one had not been previously registered which creates an *Invitation (M)* to continue using the service.

Once the user continue with the minutes of a meeting he has a very *Structured View* (M) to *Annotate* (M) the decisions and notes around the meeting, these options have a great effect in term of *Reciprocity* (D) because other users can see the work done since the application store the work done based on the initials of the name of each one of the collaborator.

Minute taker Juan Wendeus (JW) Attendees Juan Wendeus (JW). Type Note Owner Type Note Owner Decision Initials Date INFO Something Here Initials Date TODO Note Initials Date	Thesis Project My Meeti August 8 2013 GM	ng		minutes.lo
Type Note Owner Due fgdgdaf TODO Some desicions Initials Date DECISION Another Discussion Initials Date INF-O Something Here Initials Date TODO Note Initials Date	Minute taker Attendees	Juan Wendeu: Juan Wendeu:	(JW) (JW).	
TODO Some desicions Initials Date DECISION Another Discussion Initials Date INF-O Something Here Initials Date TODO Note Initials Date		Туре	Note	othen (CC) About this meeting Owner Due
TODO Note Initials Date	fgdgdsf	TODO DECISION INFO	Some desicions Another Discussion Something Here	Initials Date Initials Date Initials Date
			Note	Initials Date

Screen 003 – The service in action

TODO			Initials	Date
TODO	Note		Initials	Date
		<pre>> jump: alf + arrow keys move: alf + shiff + vp/down inserf: alf + enfer (+ shiff) delete: alf + shiff + backspace</pre>		

Screen 004 - Shortcuts

People can appreciate the work done and also *Showing-off* (D) about of their work (See Screen 003). Finally, some details are very interesting in this example. For instance, the service offers an "always available" sheet offering the shortcut keys to perform operation, this function is *At Hand* (M) for the user simplifying and improving his work (See Screen 004)

Note: Notice that the patterns previously describe here; next to them have a (A), (D) or (M) pointing that are Aesthetics, dynamics or Mechanics.

4. Methods

4.1. Literature Search

A literature search is a systematic search of published works in order to find what is already known about an intended research topic (Robinson & Reed, 1998). This method is an important part of the research process. Here the researcher can find a good foundation for his/her investigation.

As Christopher Hart formulate in his book (1998) to achieve a good literature search is important to define a topic, formulating good questions that help to understand the aim of the investigation. Then choosing keyword that will help to define the project, especially if the research is looking for a new concept; it makes necessary the use of alternative keywords to define possible solutions already done but known with a different name. Additionally, a good system to organize and record the results of the search is valuable. After a long research, a huge amount of information could be accumulated. So, a good system could be the difference between order or chaos in the research.

4.2. Pattern-based design Approach

The design patterns approach has its beginnings in the late 1970s by Christopher Alexander in his book "A Pattern Language" (1977). His focus is directly related with architecture, looking to solve regular problems in urban architecture. Then years later this approach was adapted to software engineering by (Gamma, et al., 1994). Additionally, this approach has been used also in Interaction design (Borchers, 2001) and Game Play design (Bjork & Holopainen, 2005).

So, a Pattern-Based design approach is a method that provides a hierarchical structure based on description of common issues and proven solutions to a recurring design problem (Borchers, 2001). The structure of the pattern's description contains a name, which is the name of the pattern. A description, where is describe the problem presented and a design solution to that common problem. Some authors (Bjork & Holopainen, 2005) (Bergström, et al., 2010) add some examples based on current solutions with the aim of make easier the understanding of the pattern. Additionally, relations among the patterns are a common ground when is used this method.

4.3. Iterative Design

Iterative Design is a design methodology based on a cyclic process, where is created several prototypes to analyze them in order to refine the design in many iterations (). These processes are intended to create improvements and refinement in the design, improving the quality and functionality of it.

The Iterative design usually follows the following steps; An Initial design is created as a starting point for the iterative process. This design is usually a very rough design created just to define a basic structure to work with. Then, the design is evolve several times. For each evolution notes are taking documenting issues and problems in the design to be improved in further evolutions. In every evolution refinements should be done looking for improvements and be closed to the final solution. These two last steps should be repeated until the issues documented are solved (Nielsen, 1993).

This process offers many benefits for the designer, because helps to solve misunderstanding generated between stakeholders before a market release. Encourage users' feedback, and reduce the workload in the final implementation simplifying and clarifying issues before its development (Mantei & Torey, 1988).

4.4. Prototyping

A prototype is an early model of a service, product or any design imaginable. This method is based on the concept of test and trial ideas in order to be enhanced in further iterations of the design process. This model could go from low-fidelity prototypes to high fidelity. This is based in the quality imposed in the design. Usually prototypes test ideas, which from order method could be impossible to see and understand its availability to be done or not in real life (Smith, 1991).

A prototype plays an important role in the design of a product; quoting Donald Schön (1983); "Is the activity of building prototypes encourages reflection in design" so, having said that the importance of the prototype is to encourage the discussion and reflections directly in the designer and in the stakeholders.

4.5. Personas

Persona is a method that was presented by Angus Jenkinson (1994), in his article "Beyond Segmentation" where is created a fictional character to represent customer segments and general requirements that could most of the possible population intended.

This "Persona" is a way to humanize analytical data that helps designer to connect needs and motivates certain decisions in the design.

Personas as a design tool offers valuable strengths because they help the designer to determine how the product should behave and what kind of communication should deliver for the different stakeholders. Additionally, helps to generate consensus in the design standardizing the languages and creating a common understanding. On the

other hand this method also offers a measure of the design's effectiveness, taking a look into the user needs and what is valuable or not for him/her (Cooper, et al., 2007).

4.6. Use Cases

A use case is a list of steps, typically describe in a story that define interactions between the different users of an application. The aim of this is to describe how a system works and how the users can achieve the purpose of the application through the different interactions. The use cases can describe not only human interaction but also can describe interaction between machines and desired automatic actions done by the program.

The first person to formulate this descriptive model in software engineering was Ivar Jacobson in his co-authored book Object Oriented Software Engineering (1992). In this book helped to popularize this technique among software developers. Originally the term used was "Usage Scenarios" or "Usage Case" which was taken originally from the Swedish translation "*användningsfall*". Later this term was modified to "use case" in order to make it more natural in English³.

4.7. Target Users

Originally this term is known as a Target Audience, which comes from the advertising jargon making reference to the specific group of people within a target market at which a product is aimed at (Ries, et al., 2000) target user tend to group desirable qualities of a person describing a common ideal audience, which could result interesting to exposed to an the product in development.

³ Use cases, ten years later: The history of use cases (<u>http://goo.gl/tlrzNq</u>)

5. Process

This thesis project was carried in three main processes. The first step was carried through the literature search, which resulted in a collection of design patterns useful for the purposes of this project. Then, after creating the collection of patterns, again using a literature search, it became necessary to investigate a method to help organize this collection in order to assign an order and hierarchies within the design process. And finally, a prototype was design in order to test the work done in a design environment fulfilling the aim of this project.

5.1. Patterns

In the results section of this document are presented the collection of patterns gathered during the research session. These patterns are divided in two groups: Viral patterns and Collaboration patterns. However, since the scope of this thesis was more related to Virality and Viral patterns; the Collaboration patterns were not fully developed. For that reason they are mentioned here but the description of this collection will be placed in the <u>Appendix B</u> of this document. Below they will be listed with a short description of them.

5.1.1. Viral Patterns

- a. **Simplified:** this pattern refers to the idea to make simple an understandable design (Heath & Heath, 2007) (Wodtke, 2009).
- b. Frictionless: this pattern is related with the idea to remove obstacles during the use to make easy and pleasant the experience for user using a product or service (Wodtke, 2009).
- c. At Hand: This pattern talks about the efficiency path that user have to travel in order to achieve certain goal. In other words the simplicity that users have to access rapidly the most used tools (Wodtke, 2009) (Cooper, et al., 2007).
- d. **Rippling Actions:** is related with the idea that Micro level Interactions can affect creating a macro level phenomenon (Goldenberg, et al., 2001) (Fogg, 2003).
- e. **Targeted:** not all the users are equal, that is why designs should be directed to specific group, in order to achieve the maximum reach (Wodtke, 2009).
- f. **Self-promoted:** in order to make visible a product/service this one has to contain the tools to promote itself (Godin, 2000) (Wodtke, 2009).
- g. **Reciprocity:** It is a diffuse and unstated responsibility to repay a piece of useful information shared (Kollock & Smith, 1999).
- h. **Building a Reputation:** this pattern is linked within the social structure mankind is involved where the reputation becomes to be a value of trustiness and reliability (Crumlish & Malone, 2009).

- i. **Stackability:** Viral services, relies in the possibility to spread the word as much people as possible, that is why is important to integrate and connect with other systems in order to reach a maximum universe of users (Pennenberg, 2009).
- j. **Showing –Off:** This pattern occurs when the product let the user looks better with others (Anderson, 2012).
- k. **Complaining:** this pattern brings the idea to establish a common enemy to create an easy topic for a conversation (Anderson, 2012).
- 1. **Seductively:** this pattern happens when a person is seduced by a compelling characteristic of a product, pushing it to use it one more time (Anderson, 2011).
- m. **Invitation:** this pattern occurs when a collaborator invites to other to work along in a project (Stewart, 2008).

5.1.2. Collaborative Patterns

- a. **Annotate:** this pattern allows collaborator to call back thoughts or ideas in an understandable way (Iacob, 2011).
- b. **Structured View:** This pattern refers to the idea to present information in a structured way, letting the possibility to have a holistic view of the document shared.
- c. **Collective Intelligence:** is the concept where the group as "unified intelligence" can create better solutions than the individuals.
- d. **Peering:** in the concept of establish connections with others, related by common topic or tasks.
- e. **Folksonomy:** Refers to the classification system based on tags, assigned by the same person that creates the content tagged (Stewart, 2008).
- f. **Socialize:** this pattern refers to the mechanism used by people to start connections and relations with other collaborators.
- g. **Recognition:** this pattern indicates the use of tools that help to recognize the value added by contributors (Stewart, 2008).
- h. **Kick-off:** This pattern refers with the first introduction offered in a collaboration environment to give a welcome and introduce some features of the system to the user.
- i. **Communication:** This pattern refers with the options that a collaborative tool offers to support communication among the users.
- j. **Hierarchy:** It addresses the mechanisms that enable hierarchy in the content of an application (de Moor, 2006).
- k. **Privacy:** refers to the capacity that a collaborative application has to provide areas in the document not available to others collaborators (Iacob, 2011).
- 1. **Preview:** this pattern refers to the possibility that have other collaborator to visualize what he/she has done (Schummer , 2004).
- m. **Helping:** this pattern refers to the possibility that offers a system to help users in complicated situation within the collaborative environment (Schummer , 2004).
- n. **Indexing:** this is based in grouping and organizing the content created, facilitating the later search and accessing of it (Schummer , 2004).

- o. **Structured:** this refers to the possibility to have templates in order to create predefine documents to simplify the workflow (Stewart, 2008).
- p. **Scheduling:** in order to synchronize people's agenda is necessary to book and arrange a common time to work (Stewart, 2008).
- q. **Assigning:** collaboration relies in the responsibility of others to fulfill the aim the final purpose, that is why people is assign to fulfill certain tasks (Schummer , 2004).
- r. Linked Information: this pattern refers with the need to connect content and determine threads of information in a collaborative process (Stewart, 2008).
- s. **Team-Up:** this pattern appears when people need to share content with each other and groups, so they create teams to connect and shared information easily and efficiently (Stewart, 2008).
- t. **Versioning:** This pattern shows the evolution of the content developed by the collaborators, letting have several versions in order to have easy access in case of mistake or error in the newest versions (Iacob, 2011).
- u. **Consensus:** this pattern occurs when the group is provided with tools to agree in a topic discussed during a collaborative session (Schummer , 2004).
- v. **Sessions:** this is related with the mechanisms that let the group preserve tis current state in the collaborative process when this is interrupted by any factor (Schummer, 2004).

5.2. Using and classifying Patterns

During the research it was noticeable the nonexistence of connectors around the patterns, producing a non-sense collection of design patterns, with no structure and coherence to be applied in a design process.

After some workshops carried with the thesis supervisor it came up, a cohesive model that helps to create a consistent path to follow in the development of a design guideline. This model is called "MDA model", which provides the essential tools to organize the patterns based in their type, creating relations and hierarchies to be used as solution for design issues while is thinking in virality and stickiness.

5.2.1. Using MDA Model

After having identified this model the next step was generate an understanding of the patterns collection related with this model looking for connections and placement for the patterns. Here viral patterns and collaborative patterns were combined merging them in just one group looking to have a more integrated perspective to fulfill the aim of the project. For this was done a workshop with the thesis supervisor, where it was discussed the position, level of the patterns and how they are going to affect each other (Figure 001).



Figure 001 – Workshop, using the MDA model to arrange the pattern collection merging Viral and Collaborative patterns in one group. Note how the arrows are establishing connections and relationships within the patterns.

In this process were combined the *viral patterns* and the *collaborative patterns* in the same spot. In order to create a common ground to design thinking in virality but having on minds a collaborative solution. Consequently, it was use a board divided in three areas (Aesthetics, Dynamics and Mechanics) and all the patterns were written in post-its to move them back and forward to finally find a good connection.

First the workshop started working with the *aesthetics;* in this group were place the patterns that works as a vision or concept useful as a measure to define the track the design want to take by the designer. Then the dynamics, which are the processes, actions or behaviors intended into the design supported by the functionalities defined in the mechanics. And mechanics are those patterns that include the rules to support the dynamics and fulfill the vision proposed in the aesthetical level. So, as it is shown in the table 001 the patterns were arranged.

However, these patterns are not isolated entities. Actually, they present connections and relationships with other patterns. As is shown in the Table 001, it was explored this relations and connection defining an interesting flow in the design process, creating additionally hierarchies and dependencies in the patterns.

Aesthetics	Dynamics	Mechanics
 Simplified (V) Frictionless (V) Rippling Actions (V) Irresistible (V) 	 Self-Promoted (V) Reciprocity (V) Reputation (V) Showing-off (V) Complaining (V) Collective Intelligence (C) Peering (C) Socialize (C) Privacy (C) Team-Up (C) 	 At Hand (V) Stackability (V) Annotate (C) Structured View (C) Acknowledge Goodness (C) Self-Promoted (V) Kick-off (C) Supporting Communication (C) Hierarchy (C) Preview (C) Indexing (C) Structured View (C) Template Content (C) Scheduling (C) Assigning (C) Hyperlinking (C) Versioning (C) Consensus (C) Sessions (C) Invitation (V)

Table 001 – (V) makes reference to Viral Patterns and (C) to Collaborative patterns.

The collection of patterns presented above is correlated based in their hierarchical location. So, in the aesthetical level *Simplified, Frictionless, Rippling Actions* and *Irresistible* are located creating two groups chained to build up the next groups of patterns based on these aesthetical ones. Then the process continues with the dynamics that are the actions intended from the aesthetical level that finally will be fulfilled with the mechanics proposed in the patterns collection. In other words, this relation initialized using the aesthetical patterns as the vision of the design desired, aiming the dynamics, which are the actions that should happen if the mechanics are offering the right solution to solve those intentions.

5.3. Design Process

After an interesting and exciting research process, it was time to try to put in practice all that knowledge acquired. For this design process was selected a series of workshops as starting point in order to gather information from the stakeholders (*Interaktionsbyrån*) like requirements, desires and visions related with the creation of the application in order to have a base to apply the design patterns and the design guidelines described in the results of this document.

Once a ground was established several iterations were done. Starting from lowfidelity prototypes based on sketches and schemas to finalizing with a high-fidelity but nonfunctional prototype that shows the interface of the application plus a workflow that explains how users interact with the application enabling viral actions that could release a viral chain among the users.

5.3.1. The Application

Let's write it (As is called by the stakeholders) is an application that should encourage people to have meaningful meeting discussions. Thus, based on this statement and the original aim intended; The concept to create an application that contains the entire elements usually utilized in a regular meeting like a *keynote*, *task manager* and GTD^4 application integrate to create a simple way that help people to conduct, organized a meeting and at the end of it help them to take decisions.

Many examples were taken into account taken in a benchmarking done, such as, *Microsoft Power Point, Apple Keynote, IA writer*⁵ and *Prezi*⁶. They help as inspiration to determine certain requirements in the solution. For this purposed the application was defined as a linear solution that should contain all the information necessary to have an holistic view of the meeting and have a quick understanding of what is happening on it. Then, is possible to say that this application to enable meaningful discussions. It could be called from now on an "AMD" (Application for meeting discussions). The need for a new name comes from the idea to bring with a new name for a new concept not seen before. Seemed like what happened with "*GTD*" a new application concept that born from a management concept and then evolve to whole new family of application that allows to follow the parameter established by the management concept (Allen, 2001).

5.3.2. User Scenario, Target Users

Target Users:

Originally was intended make use of personas in the project in order to have an accurate selection of the focused group, following the stickiness principles. However, it was not possible count with this information since this should be gathered with extensive descriptions and details that are out of the thesis scope. However, personas were replaced by target users in order to continue with the flow of the design as is proposed in the design guidelines placed in the results of this document.

Robert Magnusson (Meeting Creator) He is an experienced Graphic designer that works in a web design agency. A web project has already started and he needs the help of his co-workers to design this new web application. This requires a lot of collaboration to find the right mood and style for the website and its functionalities.

⁴ GTD: Getting Things Done

⁵ IA Writer: <u>http://www.iawriter.com/</u>

⁶ Prezi: <u>http://www.prezi.com</u>

Karin Rogers (Regular User) She works with Robert, she is a designer too, and she loves to work with digital graphics specially animation. She is subscribed to many channels in YouTube and has many friends in Instagram. She is always willing to help and he has a really well understanding of design software.

Emilio Perez (Regular User) He is the new guy of the company, he has the latest gadgets of the market. He is a Product designer that knows very well how to work and collaborate with others over Internet. He knows and has experience with technical stuff.

Joan Henson (Guest User) she is the client, she has a company that helps people in remote places to take care of their health through mobile devices. She is a very strict person with a very tight schedule. She likes to have concrete and concise information when she starts working.

Users Scenario:

The scenario is a regular meeting where the different stakeholders are preparing the first stage of the design creation and conceptualization of a new product. They are looking to create a brand new application for healthcare that use the mobile phone as mean to get health information from the user using the different sensor of the telephone.

Robert starts planning the meeting and asks to Karin for help to complete the agenda and look for information that provides a good kick-off inspiring other designer to work with this project.

In that moment Emilio is invited to the meeting as a support for the technical questions and requirements to elaborate in the application.

Robert also invites Joan to the meeting, her contributions are very important and she is a key factor in this design process to have a successful application. She is the main stakeholder but also she knows better that market than the designers and she can contribute with information valuable for them.

Everybody has received a message with an invitation for the meeting, showing the place, time and a general preview of what is going to be about this meeting.

Now, Robert is preparing the content of the meeting in order to have all the details ready, he is adding the agenda, some titles connected with the different elements of the agenda and some words explaining the content of the agenda.

During the meeting, Robert continues adding content assigning tasks based on the outcomes of the meeting, meanwhile Emilio is uploading some photos to create a mood board to inspire the other designers and guests to collaborate and contribute with the meeting. Finally, when all the information is done and the meeting has concluded all the task, polls, schedules and images are saved and arranged to the easy

access of other stakeholder and the same participants of the meeting to follow back their tasks and outcomes of the meeting.

5.3.3. Design Iterations

During the workshops carried with the design team of Interaktionsbyrån, making use of iterative design method, it was noticeable in the meetings the use of specific tools, to carry on and take control of the meeting. For instance, a keynote (In this case was *Microsoft Power-Point*) was used to present the meeting agenda to the participants. So, using this example as inspiration, the design process starts here. Additionally, It is stated a model where the agenda is the main driver of the meeting. Basically it is the information that should conduct the meeting providing hierarchical organization. These decisions were taken by me but counting with the collaboration of the stakeholders to proceed with the design process.

But the agenda is not the only driver; it was necessary to think about in other elements like a time schedule, task manager, annotation tools, messaging systems and others. Actually, it was clear that these tools required some kind of connection that not only serves as an integrator but also it was necessary think in terms of virality, how those functionalities could be drivers for virality and stickiness.

So, having that on mind, the approach was to explore different functions and push them to the limit in order to find solutions that can coexist together based on the viral and collaborative patterns described in this document.



Figure 002- First Iteration, accordion style to unwrap the meeting agenda and its child.

First Iteration

One of the first approaches was considering some kind of accordion solution advocating for simplified and coming with this concept of tell a story to the user (See Figure 002). This concept had in its first level the keynote of the meeting, then in the second level sub-topics and in a third level, all the possible notes, and functions like to-do and calendar. The problem with this design was that some of the most important aspects like the information generated in the meeting and the possible viral drivers were hidden, so basically the user has to unwrap the whole design before discover the whole content. At the end this is kind of similar of what it is available in the market with the keynotes application, a linear flow that hides the next information closing the possibility for the user to have a holistic view to understand the meeting and have actions and decisions on it. Thus, at the end this design was discarded.

Second Iteration

In a second analysis, a timeline path usually follows a meeting. In that sense, time as an element to generate order could be used as a design driver. Therefore, the next approach was to create a timeline, based on days where the meeting was placed in each day. The meeting follows an hourly arrangement (See Figure 003). Again the design lacked enough visualization for the user to understand the meeting, the whole flow and the information, which generated it. The problem of this is that required some kind of unwrapping to deploy the information, creating first a complicated flow that goes against *simplicity*; which it is not consequent with *At Hand* and provides friction everywhere complicating things for the final user.



Figure 003 – Second Design Iteration, exploring time as a design driver.

Third Iteration

Then a new approach was taken; an icicle tree visualization model (Chevalier, et al., 2007) came to the research as a possible solution for the development of the content. The advantages of this data structure is that allows to get an overview of the data structure, the amount of information for every single topic of the meeting and provide guidance of where are the hot topics of the meeting (See figure 004).

In this iteration of the design, the visualization provides a concrete flow of information where the hierarchy of the content is really important, generating an easy understanding of the origin of the information. At this point the design had a clear structure that could be used as way to integrate the content in a neat and understandable way for the user. However, this design was in a very early stage, since this was only a solution for the organization info. But, what is about the virality and collaboration patterns? How are they affecting the design and the actions on it?

After this point a series of meetings were carried in joint with the NIB team, in order to look for concepts that could help to solve the challenging task that represents applied the viral patterns in the design enabling virality and collaboration at the same time. For that reason, it was important to create clear functions based on the mechanics-patterns previously proposed to solve this issue. Consequently, the next step in this design process was the creation of these functions. And the refinement of them in order to find the right group that really help for the purposes of the application and worked align with the viral and collaborative patterns.



Figure 004 – Icicle tree approach that is a layered diagram that uses alignment and adjacency to organize hierarchical elements.

During one workshop the discussion was centered in which functions will be more useful for drive virality in a collaborative environment and how *simplicity* and *frictionless* could be constant in the graphic user interface. The result of this workshop in a general consensus of the participants, results in the following 6 functions:

- a. **Annotate:** This function refers to the ability of the users to write down content directly into the interface, to save their thoughts and ideas in a general place where invitees of the meeting can have access. This function follows the collaborative pattern that has the same name and also is align with *seductively* and *reciprocity* because of that idea to repay with valuable content for the people using the application.
- b. Share: When is talked about sharing, it becomes a series of thoughts about the broadness of this concept, opening endless options to fulfill this aim. So, for the purposes of this project, share means the capacity that it has the application to share, through email, social network or other messaging channels about; changes, information, Tasks, conclusions, votes and/or new content generated into the system. This function supports patterns like *Consensus, Scheduling, At hand, Stackbility, Reputation, Collective intelligence* and *Team-up*.
- c. **To Do:** This function is very significant in the relation of the collaborative and Viral patterns due to is the main driver of the meeting that finally is going to help drive virality because everybody involved in the meeting will be trying to finish the task imposed. To do, is a function that let people assign to others, tasks related with the meeting by itself or tasks related with the outcomes of the meeting. Additionally, users can assign to themselves or to other users. The concept come from the idea to engage the users pushing them to fulfill the task assigned by them or by others. This function supports the following patterns: *supporting communication, indexing, structured view, template content, scheduling, assigning, hyperlinking, reciprocity, team-up, Rippling Actions, targeted, at hand and Stackability.*
- d. **Invite:** When someone is invited to the system this is triggered by every single possible action into the system. So, for example when a user assign a new task or ask a question to someone that is not included in the invitees list this person in immediately invited to the system through email or some messaging system to connect the person with the meeting and transforming the system into a sticky application that attract people a hold it as much as possible to show him/her the benefits of the concept. Besides this concept, of course the system should provide more conventional ways to invite users, similar like what is available in a regular mail client. For instance, a field bar where the names of the invitees are added. This function is directly connected with the stickiness rules.
- e. Vote: during the workshop, most of the functions were voted in order to select the best ones. That was a useful and successful way to generate *consensus* among the participants. So, it felt natural to have this as a regular function in the application. What is intended with this function is create a mean for the users to ask question, before, during and after the meeting about issues

concerning with every single aspect of it. Definitely, the most obvious pattern for this function is *consensus*, which at the end is a great driver for Virality and collaboration enhancing the *reputation* of the users and *rippling actions* to others.

f. Attach: This refers to the possibility to attach or upload multimedia files into the content generated by the application. For instance, the meeting group is discussing about a design topic. So, they need to place some images in content to create a mood-board to inspire new designs and new concepts. The way this is done engage the users to create an *impactful* approach and to be seduced *one more time* by the application.

At this moment of the design structure was clear, the functions, the user scenario and the general concept was arranged. Now, it was time to arrange some kind of language in the design that helps to organize the information to drive collaboration and virality in the workflow. In an initial state was intended utilize a textual translation of the icicle tree to the interface but this creates a cluttered and chaotic interface making it too messy and complicated to used creating a difficult system that it was not aligned with the idea of simplicity, impactful and frictionless (See Figure 005).

Then it was taken a different approach were icicle it was remaining. But, it was added columns to separate the content into three different levels:

- First level, have the meeting agenda
- Second level, have an enhanced view for the viral drivers
- Third level contains the content shared and all the action to collaborate and enable teamwork. Additionally to this the viral actions were available here too and represented in the second level.

Now, as is mentioned above, the design is structure in a common language that provides logical flow for the user. This kind of understanding simplifies the work with the application and enables the possibility to fulfill several aims in just one system as it was intended at the beginning of the project (See figure 006 and 007).

The design has a solid structure to be used as base for a high-end prototype and define the workflow of the whole system. This workflow is going to be explained in the design rationale and there it will be an understanding of how the functions are enabling some of the dynamics listed in this document. Is important to understand also that not all the patterns were included into the final design; this was because some of the patterns could create clashes with other blocking the system flow. So, for the sake of the main purpose those patterns were not included.

Welcome to the Meeting	Hi, Welcome! Long learned for all and, consection adjuicing all, set de element tempor incidites tal labore et delare magna alique. Ut erim ad reim veniam, qué notitual exercitation capitate non poietre, sur la capita d'itela desent molt arin d'erit "Temportander la voluptée will esse citum doire eu lugier nule partaux: Ecospteur aint occaecat applicate non poietre. Sur la capita d'itela desent molt arin d'erit "Temportander la voluptée vell esse citum doire eu lugier nule partaux: Ecospteur aint occaecat Summary Previous Long learne doir sit anné, consecteur adjuision all, sed de element indicate une duir n'experimenter in voluptée vell esse citum doire eu lugier nule partaux. Ecospteur aint occaecat capitatien monoist. set in cuba qui office desent molt arin d'erit "Temportander in voluptée vell esse citum doire eu lugier nule parteux. Ecospteur aint occaecat capitatien monoist. set in cuba qui office desent molt arin d'erit "Temportander in voluptée vell esse citum doire eu lugier nule parteur. Ecospteur aint occaecat capitatien monoist. set in cuba qui office desent molt arin d'erit "Temportander in voluptée vell esse citum doire eu lugier nule parteur. Ecospteur aint occaecat capitatien monoist. set in cuba qui office desent molt arin d'erit "Temportander in voluptée vell esse citum doire eu lugier nule parteur. Ecospteur aint occaecat capitatien monoist. set in cuba qui office desent molt arin d'erit "Temportander in voluptée vell esse citum doire eu lugier nule parteur. Ecospteur aint occaecat capitatien monoist. set in cuba qui office desent molt arin d'erit "Temportander in voluptee velle esse citum doire eu lugier nule parteur. Ecospteur aint occaecat capitatien monoist. set in cuba qui office desent molt arin d'erit "Temportander in voluptee velle esse citum doire eu lugier nule esse citum doire eu l	
Application design meeting; Brainstorm, Co-Design	Brainstorm! Long the same door at answ, consection adjusting will, and do alterned tempor holdshift of bloom of door magna aligas. Ut anim ad minim version, gain notiful association of consection and provider, and in colorado and provider in colorado and provider in the colora	
Meeting Results and Outcomes	Results! Results! Consistent and, consistent and, consistent a depisation of the element denomination of the barrier of barrier and a data. Ut even and element and particle description of the comparison of th	

Figure 005 – Direct translation of the icicle paradigm



Figure 006 -

, survey of soft, and are and a	
HEADER IA Toda a Teat	
Sandary - E	
finge-preview 2	
HEADER 2 TODO D	
calm D	
Invitors	
("pre-icu text)	
Acada 3	
Vical	

Figure 007 -

After gather a solid structure it was necessary to proceed with a little bit more ended prototypes in order to find the design style and take a look for a more realistic prototype.

Welcome to the Meeting	Welcome! Lene (spen dotr st and, consected adjuscing wit, sed do examed tempor inclidant ultabore adjusce magna adjus. Previous Meeting Lene (spen dots) Lene (spen dots) at and, considering adjusce (spen dots) State State
Application design meeting; Brainstorm, Co-Design	Breinstorm Lessen and a second at a definiting with well do the found the tempor Definition of the temporary attracts. Selected pieces Co-Design Less 10
Meeting Results and Outcomes	Bressetts Gramma data meth, consected ar adjoising sit, set do suitand temps Gramma data meth, consected ar adjoising sit, set do suitand temps Gramma data meth, consected ar adjoising sit, set do suitand temps Gramma data meth, consected ar adjoising sit, set do suitand temps Other methods Breast consected ar adjoising sit, set do suitand temps Other methods Other methods

Figure 007 a- Digital Prototype



Figure 007 b- 4K prototype including functions and viral drivers

his is a Header	Welcome to the meeting
	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut al- iquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
	url:
	Browse
	Done
	+ > Picture Collection Video add todo Pol

Figure 007c - Low-fi Sketch showing interactions and working modes



Figure 007 d-Digital Prototype Including refined functions and interaction modals

The design have evolved from a basic digital dumps (see figure 007a) which provides a preview of the general content giving in that way a glimpse of the meeting however this concepts is still difficult to understand since the user have to learn how it works this if he/she wants to use it.

Then, as a matter of experimentation, one of the iterations explore the possibility of Ultra HD definitions (3840 x 2160 pixels) with the aim to fit as much information as possible looking for a futuristic approach where this kind of definition will be the standard (see figure 007b). In this prototype is possible to appreciate the that all the content of the meeting is intended to appear, and the user just need to do a zoom in in order to check the details. However, this approach makes a problem for the current screens resolution so was discarded. Additionally, it was remained the problem of how people create content for the meeting. There was no an intuitive way to understand how the content should be created.

The first approach taken to solve this issue was to create edit mode where users can interact to create the content. In this mode the user has a menu bar to access to the most common functionalities. However, this solution is still complicated since the user has to discover this functionality because is hidden in the no-edit mode (See figure 007c). Then in a previous iteration to the final design, a more visible option was created (See figure 007d). Here, the user is provided with tool located in the edges of the screen to create new content. Like agenda items and functions like share, polls, etc. the problem of this, results in a tedious and boring way to interact. For that reason it was necessary to come up with something surprising but still easy to use. Next in the results it will be explained the final solution of the design.
6. Results

The results presented in this section are divided in two groups. Firstly, the collection of patterns and how they should be applied in a design process in a form of guidelines. Secondly, the design of an application, where is applied the patterns and the design guidelines previously mentioned.

6.1. Stickiness Rules

The stickiness rules are not directly a result of the research. However, they are included in the results, because they act as meta-patterns for all the viral patterns and collaborative patterns. This means that they should be considered all the time when is designed for virality. The <u>Stickiness rules</u> could be found in the theory section.

6.2. Viral Patterns

This collection of viral patterns is part of the literature search. As a result this collection is made from information found in various sources. Here, It was made an effort to describe them as patterns, exemplifying and organizing within the MDA model defining a structure and hierarchy.

Below the patterns will be presented based on the MDA model, starting with the aesthetics and followed by the dynamics and ending with mechanic patterns.

Simplified

Although this looks obvious, it is worth making a strong emphasis on this principle using it also as a viral design pattern. From time to time designers tend to become fascinated with finer extra details, but it is easy obfuscated and lost the message and meaning of the design, adding more than necessary. According to the Heath & Heath (2007) quote:

"... You need to keep your message "succinct enough to be sticky, meaningful enough to make a difference." Analogies often make things simple and easy to understand because you're comparing new concepts to something they already know..."

Having this on mind, the idea is it possible to argue that simplified could be based on the creation of simple designs easily understandable that relies in common concepts and simple behaviors where less is more (Cooper, et al., 2007).

Example 1: When a user has installed in his computer the Dropbox client, this user is allowed to have a copy of their local files in the cloud. The simplicity of the service is just copy and paste in a specific folder and the system is in

charge of the uploading, management of the files and versioning. So, no technical knowledge is required! What makes this interesting is the notion of easiness; because the user just relies on his regular knowledge of copy and paste. The rest of the interaction and functionality is driven by the system. Thanks to this simplicity the user is pleased and provides a positive feedback becoming good for the Word Of Mouth (WOM) between the user and people who share their files.

Example 2: Instagram is a great example of what simplicity means. Users can upload a photo and apply some predefined filter(s) that help to improve the quality of the photo, transforming them from normal photos to stunning images with incredible color and awesome effects. Compared with the time that something similar could take in a program like Photoshop this process is simple and does thus not demand so much effort from the user.

Using the Pattern: The primary design choice when design is based on *simplified* is to analyze the design and check what is necessary and what not, iterating in this process several times until the design provide only the necessary elements on place. Usually this process requires a constant iteration in other to test the value of certain functionalities to find the right ones in the right moment, creating a very understandable solution for the user.

MDA Position: Simplified is placed in the Aesthetical level

Frictionless (Effortless)

This pattern is about to remove any doubt to the user in order to trigger an action (Wodtke, 2009). This pattern usually occurs when users are asked to activate, sign up or do some action into the system that requires certain time and effort. Hence, this pattern is about the way a service or a system provides an effortless method for the user in order to simplify the adoption and use of it.

Example1: Spotify makes use of Facebook Connect⁷ in order to simplify the Sign in / Sign up to start using that service. In that way what Spotify is doing is simplifying this process for the user having a better adoption rate and reducing the friction generated by the filling of a long and tedious form. In other hand this pattern creates a win/win for both parts where Spotify can get information from contacts and personal data and the user can use the service just with one click action.

⁷ Facebook Connect: <u>https://developers.facebook.com/docs/concepts/login/</u>

Example2: Another example of *frictionless* is the procedure performed by Facebook, leaving the last configuration of the user as the default⁸. With this action Facebook has *simplified* the steps for every single user; instead to continuously ask for changes in the privacy settings. Of course there is a backup; people can change always their privacy settings. However, the easy method to reduce friction is having a default value; then users can do what they want.

Using the Pattern: When an is designed for *Frictionless*, the design must be *simplified* in order to eliminate any barrier that require some tedious, boring or complicated process that could make the user lose his motivation to use the service or product. At the beginning it has to be taken into account to create a *simplified* adoption process to acquire information about the user. Then, in other steps that require complicated settings the system should come with default settings to make them easy for the users. If some function is too complicated, the design will be like a wall preventing a good flow in the interaction making the user feel bored and frustrated to finally abandoning the service and creating a bad reference (Negative WOM) for the application or product. Effortlessness is the avoidance of excise (Cooper, et al., 2007) in the processes to be performed by the user. Too much effort means less people using the service and more people sharing negative information about the application, product or service.

MDA Position: Frictionless is placed in the Aesthetical level

Rippling Actions (Impactful)

This pattern is related with micro level interaction that affects macro level phenomena (Goldenberg, et al., 2001). In other words micro level interaction can be simple actions that have a huge impact among the mass, creating a ripple effect that influences many, maximizing the reach of that action. Additionally, is possible to argue that an impactful action is commonly related with a motivational value that triggers an action (Fogg, 2003).

Example 1: YouTube has a function where people can like or dislike a video. This function is visible for everybody that visits the video. Thus, when a person push the like/dislike button is creating a micro level interaction that in combination with others creates a trending popularizing the video, bringing the meme phenomena. YouTube is plenty of these examples. For instance, the video-music called "Gangnam Style" performed by the Korea artist "Psy" which in less than 6 months becomes very popular going from 1 to 1000 million of views and 5 million of likes.

⁸ Facebook default privacy: <u>https://www.facebook.com/help/260276693997558/</u>

Example 2: In 2007 Twitter implemented the hashtag⁹ (#) functionality on their service. The hashtag creates the option to track specific topics, which in turn let the user to know which is the current trend around the world or in his/her location. This small function has an impactful effect that can even help to modify internal policies on established governments¹⁰.

Using the Pattern: When is designed for *rippling actions* the designer have to think in functions that helps to maximize the user's interactions to reach a bigger group. This could be achieved using many proven solutions like hash-tags, liking, rating and voting buttons that help to "hear" in some way the voice of the mass. *Rippling actions* works connected with *At Hand*, providing a ubiquitous possibility for the users into the service or product.

MDA Position: Rippling Effects is placed in the Aesthetical level

Targeted

Not all the users are equal; some are core users and others are just peripheral users (Wodtke, 2009). Hence, this pattern is related with the creation of specific tools for specific user. This creates an affiliation and a value incentive among the core users.

Example 1: Wikipedia is an example of a targeted tool; where just a few small groups contribute¹¹ with the content and millions of viewers¹² consume this information. That is why this platform offers very specific way to create and manipulate content, disregarding common users without any knowledge of the "wiki" content system.

Example 2: Flickr is an example of a targeted application that has its focus in artist and photographers who wants to share their work. Flickr offers specific tools like tagging and image retouching to improve the classification and quality of the images. For that reason Flickr is a common place to reference photography and search for impressive images.

Using the Pattern: The first thing is trying to figure it out who is the user and what is doing. Then the application has to perform the right functions in order to provide the right experience to the right user. Additionally, the application could have several experiences that adapts to every single user. However is always significant keep in mind to have it *simplified* and *frictionless*, improving interactions for the general purpose of the application or service.

⁹ Twitter Hashtag: <u>http://en.wikipedia.org/wiki/Hashtag</u>

¹⁰ Can Twitter Change The Way Governments Act? <u>http://www.youtube.com/watch?v=031UD8Xwt8A</u>

¹¹ Wikipedia contributor statistics: <u>http://stats.wikimedia.org/EN/TablesWikipediansContributors.htm</u>

¹² Wikipedia Page-Views: <u>http://www.alexa.com/siteinfo/wikipedia.org</u>

Irresistibility (Seductively, One More Time)

This pattern occurs when a person is being seduced by a compelling characteristic of a service or product. This pattern occurs when a user is tempted for some detail or function that engages him in some sort to come back to the service again. It is that thing that seduces the user to come back one more time (Anderson, 2011)

Example 1: The use of nostalgia is a quite common example of a compelling product. For example; The latest versions of Super Mario Brothers makes is of nostalgia as a compelling pattern to design the video game bringing the old "Mario Gameplay Style¹³" of playing advocating for all those veteran players.

Example 2: Kickstarter that advocate for people in order to look for founding creative projects. People attracted to compelling project make their donations with maybe some kind of reward afterwards like for example what happened with OUYA¹⁴ a video game console based on an open source platform letting people tinkering and change configurations of the platform. This project was viral and got a lot of foundation to make it possible, actually much more of what they were expecting.

Using the Pattern: seducing users is one of the most complicated things to think when is designing for *irresistibility*. For that reason is important to analyze the target group and look for desirable features to be on it. Elements like nostalgia, familiarity or the concept like challenge, commonly applied in video games were the player is seduced to play one more time to beat the game and success with the challenge proposed. Other tools could be use like curiosity, teasing with a small bit of interesting information the user this one maybe want to know more (Anderson, 2011). In general, the most important think to take into account is look into the group the product is intended, find the most seductive and compelling factor on it and enhance them to attract the users to come back.

MDA Position: Irresistibility is placed in the Aesthetical level

Self-Promoted (Outreach)

As Godin (2000) has established that a viral service / product should promoted by itself to be visible for the people. That is why this pattern follows this concept to

¹³ Super Mario Wii U: <u>http://newsupermariobrosu.nintendo.com/gameplay/</u>

¹⁴ Ouya – Kickstarter project: <u>http://www.kickstarter.com/projects/ouya/ouya-a-new-kind-of-video-game-console</u>

provide the tools to recommend this to others. The service must provide a way to recommend itself. With this, people can corroborate the quality of the product because of its recommendation level (Wodtke, 2009).

Example 1: Amazon provides a recommendation system, which evaluate books and products based in a rating provided by several users. This creates a huge mass of knowledge that helps Amazon to create profiles based on tastes and likes. The interesting value of this is how just the contribution of a small piece of information builds an accurate recommendation system helpful for the company and the users.

Example 2: YouTube has become one of the most popular sites in the world with millions of videos store in their servers. But, when several hours of videos are uploaded daily¹⁵ it is very difficult to keep track of everything. YouTube offers "Like/dislike", "Views" and "Share" helping of memes¹⁶ to be real and visible for many. Now, people can see trends and what's going on.

Using the Pattern: Designing for Self-Promoted requires to think in smart ways to include promotions of the services in a slightly way where the information shared comes with a small piece of non-intrusive information of the origin. In this case logotypes, images and even pieces of the interfaces that resemble the origin of the shared information could work.

MDA Position: Self-Promoted is placed in the Dynamic level

Reciprocity

This pattern involves a diffuse and unstated responsibility to repay a piece of useful information shared (Kollock & Smith, 1999). This is a great motivational driver to push people to do or contribute in certain task like share or talks good about something. Thus, people who receive something are prospective payers.

Example 1: Linked-In has a specialized function to ask for recommendations to other friends or co-workers. In this function people recommend about the person's performance in their jobs. As soon as the recommender finishes the system place the recommendation into the person recommended profile. But, additionally this person is asked to "repay" sending a recommendation too. So, is quite likely that will motivate this user.

Example 2: There are many open source projects that are based on the *reciprocity*; one of them is Linux. Programmers receive a stable and solid

¹⁵ YouTube press statistics: <u>http://www.youtube.com/t/press_statistics</u>

¹⁶ Meme: an idea, behavior, style, or usage that spreads from person to person within a culture (Merriam-Webster Dictionary)

operating system to create powerful and professional servers for free. In exchange they are asked to support the cause with code and new improvements for the system with the promise of living the source code open for others to continue sharing and improving. So, the programmers based on this community are motivated by the service obtained repaying with lines of code and support for improvements.

Using the Pattern: This pattern could be used in several ways; Groups and/or Communities motivated by a bigger purpose are always useful to be implemented. Additionally collaborative environment are too raw material to explode. Basically, this pattern is about giving and getting benefits. Someone offers something and other makes use of that having the need to repay for the "favor" received.

Free basic features are a good technique to start with. However, in today's world many services already offer free features, making this as a commodity among the users. Therefore, ingenious methods should be applied. For example, gamification¹⁷, this could be used to persuade people to something in change of badges or trophies.

MDA Position: Reciprocity is placed in the Dynamic level

Building a Reputation

This pattern is related with a person that is linked within a social structure, which expects to develop certain reputation among his/her equals through his/her achievements and/or work (Crumlish & Malone, 2009).

Example 1: The reputation that someone has in Yahoo Answer is very important to validate the quality of the answers provided by others. In this way people that has a good reputation are trustable a reliable so people can believe what they say.

Example 2: eBay based its network trusting in others. Basically people buy stuff from strangers and the only way people can trust is based on the marks assign according to its selling performance. This grants some kind of reputation for the seller in order to make him/her trustable or not.

Using the Pattern: The principal choice to take when is designed for *Building a Reputation* is what kind of content people is dealing with. In that way, the designer can think if is require or not the trust of the people that manipulate the content. Additionally, the designer have to reflect about the mechanisms to judge the produce content and how and who is going to rate this content. Are going to be the editors, the users or some kind of algorithm?

¹⁷ Gamification: <u>http://en.wikipedia.org/wiki/Gamification</u>

For this there are many solutions that could be implemented like stars rating systems or algorithms that calculate the use-frequency, similar like the one used in YouTube. However, the solution to be implemented has to be visible enough to enhance the reputation of the user.

MDA Position: Building a Reputation is placed in the Dynamic level

Showing Off – (Bragging)

This pattern occurs when a service or application let the user make him or her looks better with others. This pattern is permitting the user have a self-rewarding tool. It is important to note that *showing off* selfish and altruistic porpoises could drive the pattern but always keeping the self-rewarding motivation (Anderson, 2012).

Example 1: Showing off is part of Facebook's structure; here people share everyday about their travels, the restaurants they have visited or the new stuff they have acquired. Then people like and comment about this; creating a wave of reactions from the content posted.

Example 2: Thanks to the filters offered by Instagram people can brag about their pictures and the great meal or moments they had. Definitely, this application is built on the concept that people is interested in show, share and look places, meals and many different thinks by the pleasure of showing off.

Using the Pattern: Designing for *showing-off* needs to create an irresistible mean to share with others. This is targeted within a specific group similar what it happened in Instagram that is targeted to people with a mobile device with camera. Tools that enable showing information by uploading or embedding data are welcome. Additionally if *simplified* this pattern could be used as way to have cross information from several services, supported by the *Stackability* of other systems, letting the user reach his/her data from other services to *showing off* what he/she has done with others.

MDA Position: Showing Off is placed in the Dynamic level

Complaining

Complaint is an easy icebreaker among strangers, creating a common enemy to ally against (Anderson, 2012). This pattern usually has a negative connotation. But, it is useful driver of virality and stickiness among users. Is sticky because is simple, concrete, credible and always related with stories about the bad and evil of certain

product. That is why this pattern is very important to be taken into account and a probably used wisely to motivate people to behave in certain ways.

Example 1: Popular Forums like The Verge are founded in the complaints of the users about gadgets or software applications. Someone starts complaining about a specific device and people follow him/her with comments or oppositions about his/her post resulting huge discussion of a topic.

Using the Pattern: this pattern should be used carefully, due to people that complain all the time about something can be also start complaining about the service provided. So, in that case the best solution is to provide communication systems like chats or forums where people can share their thoughts. Other option is to provide with buttons to dislike the content for example the dislike button in YouTube, which creates a interesting way to attract people with negative ideas about some content.

MDA Position: Complaining is placed in the Dynamic level

At Hand

The service should provide the most important things next to the user for their easy access (Wodtke, 2009). It is creating an efficient path that users can understand and follow the flow of the design letting them achieve easily the purpose of the service, system or product (Cooper, et al., 2007). This pattern is based on the value and *frictionless* that a person can perceive from the service or application. When an application promises to do something is expected that promise will be fulfilled by certain functions easily accessible for the user, providing value and a *Simplified* workflow.

Example 1: YouTube is a good example of *at hand*. When users are watching a video they have the "like to", "share" and "statistics" right next to him/her. They do not have to spend time looking for the tools, they do not require a complicated manual to find, understand or use the most important functionalities. Everything is at hand easy to access and use.

Example 2: Facebook news feed, as some people used to call it. There is a space asking for "what's on your mind?" the user is provided with the most important things *at hand*. As soon as any Facebooker starts using this space they are provided with a set of tool to share their location and upload photos or video to post what's on their minds.

Using the Pattern: Designing for *at hand* pattern means to analyze and figure it out what users will do more often in order to make the application useful and shareable for them. Complexity is not allowed here, every single action should be at hand to

provide the ability to trigger action aiming for the fast interaction with the system and the easiness of sharing their actions. Additionally, the design compels highlighting the tools useful for sharing aim this action to increase the "WOM" into the application.

MDA Position: At Hand is placed in the Mechanics level

Stackability

Today viral services are built up based on other services; the integration of several networks provides the capability to create a spreading flow much easier than just one by itself (Pennenberg, 2009).

Example 1: the mutualism generated by Spotify and Facebook is an example of Stackability. Spotify is built up over the connectivity network offered by Facebook. So, Spotify does not have to concern about managing passwords and profiles. Instead they are concentrated in their core leveraging in the Facebook connect service that provide all the functionalities needed.

Example 2: Dropbox does not have any physical property except by their offices all the service is built on the Amazon servers¹⁸. The service is based in a third party architecture letting them to manage and have control of that while there are concentrated in the core of the business and improve the service constantly instead of worrying in servers and infrastructure management.

Using the Pattern: Designing for this pattern is a decision that has to be taken in what kind of interconnectivity and availability is necessary for an application. Additionally is important to notice which network are valuable to spread the word. The most important thing to take into account is that the key is to concentrated in the simplicity helping the user to be connected and integrated as much as he/she require. This makes the difference of a viral application.

MDA Position: *Stackability* is placed in the Mechanic level

Invitation

When a user is invited to work along others in a project, this pattern occurs. This is the first interaction with the service or product, which create the motivations to continue using or not the product (Stewart, 2008).

¹⁸ Dropbox build in Amazon Servers: <u>https://www.dropbox.com/help/27/en</u>

Example 1: In Asana¹⁹ a task management web application, when someone is invited to participate in this person receive an email with a invitation to an specific project, the email has all the necessary information to know the deliverer and the project details. This is the first interaction a user has with the application and works as an introduction to enter into a collaborative environment.

Example 2: In Google Docs when is shared a file between a group of people. These people receive an email with details of the document, which has created and options of visualization. Additionally, show the kind of document giving a glimpse to the viewer of what he/she is going to work with.

Using the Pattern: while this pattern is very important to create a very first approach with the user it is necessary to take into account some other pattern from the virality and stickiness in order to create an effective way to attract people into the application. Otherwise, it could be taken as spam or some annoying message from a machine.

MDA Position: Invitation is placed in the Mechanic level

6.3. Collaborative Patterns

Most of the effort of this project was focused in the creation and organization of viral patterns to then continue with the final design. The collaborative Patterns were not entirely developed as the others patterns. Nonetheless, an effort was made in these patterns, which are located in the appendix of this document since they still need to be refined (See <u>Appendix B</u>). However, they are considered in the design and the integration of the MDA model.

6.4. Patterns within MDA model

The MDA model was intended original for game play design. However, in this thesis project they were used in a regular interaction Design approach. Because of this novel use, the patterns included in the MDA model are part of the results of this project. Next below, the list of the <u>patterns arranged</u> in the MDA model is listed and the details of this could be found in the process section.

Aesthetics Dynamics		Mechanics
 Simplified (V) 	 Self-Promoted (V) 	 At Hand (V)
 Frictionless (V) 	 Reciprocity (V) 	 Stackability (V)
 Rippling Actions 	 Reputation (V) 	 Annotate (C)
(V)	 Showing-off (V) 	 Structured View (C)
 Irresistible (V) 	 Complaining (V) 	 Acknowledge
	 Collective 	Goodness (C)
	Intelligence (C)	 Self-Promoted (V)

¹⁹ Asana, GTD app : <u>http://asana.com/</u>

 Peering (C) 	 Kick-off (C)
 Socialize (C) 	 Supporting
 Privacy (C) 	Communication (C)
 Team-Up (C) 	 Hierarchy (C)
	 Preview (C)
	 Indexing (C)
	 Structured View (C)
	 Template Content
	(C)
	 Scheduling (C)
	 Assigning (C)
	 Hyperlinking (C)
	 Versioning (C)
	 Consensus (C)
	 Sessions (C)
	 Invitation (V)

Table 001 – (V) makes reference to Viral Patterns and (C) to Collaborative patterns.

6.5. Design Guidelines

Something valuable to take into account during this process is figuring it out that not always all the patterns fits in every single situation. They are like building blocks, which by their own selves do nothing, but combined can be a powerful tool to design and keep focus in details while they help as a solution for common problems.

Said this; patterns are not intended to be used as a formula to solve design problems. Instead, they are useful as guidance to common design problems. Every design case requires research, analysis and a deep understanding of what is required. So, try to define a formula it is impossible; there is no solution that fits well in everything. Now, something possible is to establish a common way to starts and follow some structure to tackle a design issue thinking in virality as a design driver. This process is divided in following four steps:

First Step: Create a Story

The first step starts with a conscious exercise looking for a story. Yes a story, something that could tell us what is the product, for that is necessary begins naming in a concrete and understandable way the product. This step is very difficult and requires time and effort; this step is perfect to be done in-group that is motivated to find new approaches. As soon as is given a name to the product is good to think about how this product affects people and find an emotional connection that could be used as enhancement into the product. For example, Dropbox always appeal to safeness and ubiquity, arguing that photos and files are more secure and accessible with them. This is a compelling argument that pushes people to have an extra copy of their files, however as soon as they notice they do not have enough space are required to have more and more.

Second Step: Make it Credible

Something that is important about this story is to make it credible and concrete. This is very important to enable the "word of mouth" among the users. This is a key feature that has to be thought in order to spread the word fast among the people. Once, the story is done, this one should be tested with other people different of the design team; testing the story helps to refine the concept and make it really simple, unexpected, credible and emotional for the final users.

Once this step is done, is needed continue looking into the patterns which of them are more related with the final product. This is a constant exercise that has to be done several times. Basically the idea is to iterate through the patterns to find fits better to accomplish the purposes of the story and the aim of the product. Design for virality at end has as a purpose creates compelling products that are successful among their users.



Figure 008 – Design guidelines model.

Third Step: Find the Aesthetics

The list presented in this document is extensive. So, it's good to start looking first from the patterns grouped in the aesthetic list. These patterns evoke the desires and global concept to be implemented on it. This is an extensive process that brings to the designer a global approach to start with. For example, in the list *simplicity* is an aesthetic viral pattern that immediately fetches the need to erase from the designer's mind complex systems with intricate behaviors.

Then, the mechanics patterns are the ones to follow; those patterns help the user to make possible the ideas developed in the aesthetics list. Basically, it could be said, that the mechanics are the enablers of the aesthetics patterns, they are the ones that help to make possible the concepts of the general vision.

Forth Step: Iterate between Dynamics and Mechanics

Finally the dynamics are the final step to follow; they are the regulators and evaluators of the two previous groups. Saying in other words, if the patterns applied

can fulfill the intention proposed in the dynamics, the design is following the right track. For example, it is possible to see that the design of Minutes I/O^{20} product because it follows *simplified*, *frictionless as* Aesthetics and *Stackability / At Hand* as Mechanics patterns which in turn enables to *collective intelligence / Peering / Team-Up* that are dynamic patterns. So, is fulfilling the intention through its mechanics and dynamics.

This design approach is the one used during this project. However, how these patterns could be addressed, is totally up to the designer. It requires some time before the designer sensitize with the patterns structure and the several options available. But, something that is always truth is that the designer requires a constant iteration among the Aesthetics, Mechanics and Dynamics going back and forward, testing and looking for the optimal option to provide a the right service to the user in order to hooking him into the system to push the user to share with other in order to make it viral.

²⁰ Minutes IO: <u>http://minutes.io</u>

6.6. Final Design

After iterate in the design, finally an structure was defined and a solid structure based in the MDA model was set to be used as a ground for the final design. This application works as a collaborative tool for meeting enabling meaningful discussion among the participants. The vision around this application is to create a solution that is not dependent of a specific device. For the purpose of this project was only done a desktop version that could be used as a power board where people sat in a same meeting room is sharing in a big screen the content of the application. While, these participants can also add content in they own devices.

Next below it will be presented the final design achieved in joint with the stakeholders, explaining the application workflow and the rationale of the design.

Application Workflow

The application starts with a clean canvas divided in three sections. Header, Left column and right column (See Figure 009). The meeting creator starts in the header adding a title and the name of the initial participants (See Figure 010).



Figure 009 – Application workspaces (To take a look of those screenshot in higher resolution go to: <u>http://goo.gl/ye1E5</u>)



Figure 010 – Inclusion of title in the application

As soon as the meeting creator has written the title and has added some users the automatically invites the people added to join the meeting, they will have more information according the information is developed more. Now it is necessary to add a topic in the agenda this is possible clicking the message "click to add new agenda" (See figure 011).

EXAMPLE 2 Let's Write it, Start-Up Meeting! Monday 02/03/2013 Göteborg, Sweden	🔔 Juan Wendeus: Joel Rozada; Emma Rozada;
Welcome to the meeting	Click to add Title
	Click to add Content
Click to add New Agenda	

Figure 011 - A new agenda title has been added

With the addition of the agenda item, the user is enabled to add content related to the agenda in the right column. The minimum content possible is a title and some words in order to describe what is going to be done in that stage of the meeting. Immediately the user writes some title, this one appears again in the left column next to the agenda

(See Figure 012). The purpose of this is to create a direct relation of the agenda and the topics to treat during the meeting showing the hierarchical order of the content.



Figure 012 – "Welcome Back" is the title of the topic to be treated in the agenda "Welcome to the meeting"

Once the title is added and some words are written next to the title, the whole block of content is highlighted in white in order to match with the left column creating a general group of content. Additionally a new blank space is created down below to the previous title aiming for the addition of more content (See Figure 013).



Figure 013 – Title and Content matching the white clean user interface.

Every single piece of content is connected with the block of function previously mentioned in this document. These functions are presented below the content in form

of an accordion that it could be unfolded when is touched one of the functions (See Figure 014).



Figure 014 – Folded Function connected to the content

Then the user can choose between create a "todo", "poll", "schedule" or "Notes & *images*". When one of those is touched immediately displays a block with some fields where the user add information related with the specific function. For example in the case of "todo" the fields offered are a short note describing what is require to do, the assignees to the task and a specific date. All these field are mandatory except of the date (See figure 015).



Figure 015 - Creating a todo

$\equiv angle$ Let's Write it, Start-Up Meeting! Monday 02/03/2013 Göteborg, Sweden	🛓 Juan Wendeus; Joel Rozada; Emma Rozada;
Welcome to the meeting Welcome Back	Welcome Back Lorem insum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
	To do
	Image: Ask a question
	Write an option Write an option
	Write an option
	Add more options
	Do you want to ask this to someone in especial?
Click to add New Agenda	E For
	E Notes & Images
	Click to add Title

Figure 016 – Creating a poll

The next function available is "*Poll*", here the users are allowed to ask about anything he/she wants. In this function the user is required to write a question, some answers and the people assignee to be invited to answer the poll. If no assignees are added the poll will be asked to all the invitees to the meeting. This function let user only to select one option in the list of options created (See figure 016).

Let's Write it, Start-Up Meeting! Monday 02/03/2013 Göteborg, Sw	eden 🔺 Juan Wendeus: Joel Rozada; Emma Rozada;
Welcome to the meeting Welcome Back	Welcome Back Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
	To do
	Poll 1
	What do you want to program in the schedule?
	Date
	o line.
	Schedule
	Click to add Title
Click to add New Agenda	Click to add Content

Figure 017 - Creating a schedule

Schedule is the next function available in the folded list, this function work similar as a reminder where users can add a note describing what they should be reminded, a date and a assignee in order to assign the schedule done to someone else (See figure 017)

$ \equiv angle$ Let's Write it, Start-Up Meeting! Monday 02/03/2013 Göteborg, Sweden	A Juan Wendeus: Joel Rozada; Emma Rozada;
Welcome to the meeting Welcome Back	Welcome Back Lorem ipsum dolor st amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
	To do
	Poli Poli
	Schedule ? ·
	E Notes E Images
	Winte your notes here
	Notes & Images
	Click to add Title
	Click to add Content
Click to add New Agenda	

Figure 018 - Creating a note

The final function is "Notes & images" in this function the user has the possibility to add notes or Upload images directly to the content. In the case of the notes the user create small notes that appear in the interface as post-its (See Figure 018). In the case of the images the user can upload one or several images at the same time. When are uploaded several images those area treated as a collection of images that are going to be arranged by the system automatically fitting all of them in the space available in the interface (See figure 019).

$\left \equiv ight angle$ Let's Write it, Start-Up Meeting! Mor	iday 02/03/2013 Göteborg, Sweden	Juan Wendeus: Joel Rozada: Emma Rozada:
Welcome to the meeting Welco	ome Back	/elcome Back rem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad inim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
		Z To do //
		Notes Images Click here to choose your images
		> Notes & Images /
Click to add New Agenda		

Figure 019 - Uploading Images



Figure 020 – Application populated with content and functions

After having created some content and having activated some functions in the system the interface is populated with content like images, to do, etc. At this stage of the operation is reflected in the left column all the actions done. So, is possible to see all the title of the content and the functions applied to that content. For example, in the figure 020 the Agenda "Welcome to The Meeting" is divided in two child contents "Welcome Back" and "Previous Meeting" and these in turn have some "TODOs" and Images on it (See figure 020).



Figure 021 - Visualization mode, functions hidden

$\equiv angle$ Let's Write it, Start-Up Meetin	Ig! Monday 02/03/2013 Göteborg, Sweden	🔔 Juan Wendeus: Joel Rozada, Emma Rozada;	
Welcome to the meeting	Welcome Back Workshop with Emma and Joe Workshop with Emma and Joe Previous Meeting	Welcome Back Lorem ipsum dolor sit armet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullarnco laboris nisi ut aliquip ex ea commodo consequat. Welcome Back Welcome Bac	
Application Design Meeting, Brainstorm!	Workshop with Emma and Joel	 Previous Meeting Previous Meeting 	
	Moodboard	corem psoin doito sit a nec, consecteur aupasiong en seu do clusifica tempo incluidant a fabore e doitre magna airqua. Ot enim au minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	
	Workshop with Emma and Joel	Workshop with Emma and Joel	
	Workshop with Emma and Joel	Workshop with Emma and Joel	
Click to add New Agenda		Warm-Up! Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Image: Marc Meeting Booked! Image: Marc Meeting Booked! <th></th>	
		Moodboard Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	
		Workshop with Emma and Joel	
		Workshop with Emma and Joel	

Figure 022 - Continuous Column

When is clicked outside of the right column the functions become hidden creating a edit mode and a visualization mode to be shared and presented to the other attendees of the meeting (See Figure 021). The content placed in the right column is a continuous column of information even if is created by several agendas, the ideas is that the flow of the meeting is continuous and the agendas are only a separation to add order and hierarchies into the meeting, making easy for the attendees understand which topics are going to be treated in the meeting (see figure 022).

Rationale

Finally after several iterations, the final design has been reached aiming deliver a prototype for a collaborative application to hold a meaningful discussion, based on the viral patterns describe in this document. Below, a Design rationale will be presented to show the results of the work done and how different patterns were applied.

The design combines some concepts of current application available in the market in one; trying to explore a new paradigm in collaboration. Having said that, this concept is a combination of a keynote application like PowerPoint, a notepad, a task manager and a survey tool. These concepts will help to enable certain viral functionalities into the applications that will enable virality and collaboration by motivating the user to fulfill tasks or answer questions through the survey system. These combinations creates an interesting mix that are really valuable for collaboration and used in a neat way in joint with the viral patterns describe here a fascinating solution for meetings.

The interface was thought to be fully interactive and work independently on each one of the devices available in the meeting. So, the interface should be simple, looking to encourage users to use it and adopt it easily. That could sound familiar since *simplified* and *Frictionless* are present in the top of the general concept. Thus the application is divided in three main areas: Header, left column and right column.

EX Let's Write it, Start-Up Meeting! | Monday 02/03/2013 | Göteborg, Sweden

Figure 023 – Meeting Title and list of attendees

In the Header, the user counts with a bar that offers three functions: Access to home, Name and location of the meeting and participants of that meeting. This last function, is one of the corner stones of the application since is here in the top bar where users can be aware of who was invited into the meeting (See Figure 023). This part is supported by *Stackability*, since the user is connected to other networks like Facebook or Google plus, letting him/her have access to his/her contact simplifying connections. This area was designed taking into account a *Structured View*; offering one place to look always for the invitees, even if they were included in other section of the document they will appear in this area.

When people are added to the system, they received an *Invitation* by mail providing a *Preview* of the meeting and encouraging the user to *Team-Up* accessing the application. Those interactions flows are thought in order to drive virality in the actions taken by the users. Thus, every action helps to invite others and trigger actions that could attract other collaborator into the meeting.

The left column presents the meeting agenda and the viral drivers that are the functions planned to drive virality and collaboration making use of the viral/ collaborative patterns (See figure 024). The viral drivers are called in that way because they are constantly aiming for interaction with the user. Basically, they are mechanics that are motivating the users to trigger dynamics aimed in the design. For instance, *stackability* and *invitations* are presented in the "TODOs" and "Polls" created by the users relying in the information previously gathered from other services aiming *reciprocity, reputation, Showing-off, peering* and *collective intelligence* as dynamics triggered from those mechanics. So, the elements ordered in the left column are viral drivers because they are mechanics that help to trigger dynamics and aiming the behavior desired from the users to use the application to create a collaboration that spreads virally among the stakeholders and participants of the meeting.



Figure 024- Left Column

The viral drivers (at right of the left column) are generated dynamically while content is created in the right column. The purpose of these drivers is to highlight the elements that are creating contact with the users and elements that appeal to create an *unexpected* element evoking curiosity and a *seductively* way to enhance those elements important of the meeting.

In the right column act like a blank canvas ready to be filled with information. In this area the users are allowed to add content based on text and images and also add

interactive functions like "TO DOs", "Polls", "Scheduling reminders" and "Notes and Images" these are the mechanics of the application that provoke a the dynamics intended to finally evoke the aesthetical aspect of the application generating a viral/collaborative flow in the work held in the meeting.

The design has two states; an editable state and a view state. In the view state only appear the actions already created and in the editable state are possible to add as much action as is required. The design in the view state, tries to be as simple as possible defined by the hierarchy of the actions (See figure 025).



Figure 025 - Right Column in visualization mode

Although in the edit view, the system adopts a fold style to encourage user to click and add as much actions are required, this is a playful way to make it *seductively*, always *at hand* and good for *rippling actions* (See figure 026).



Figure 026 – Right column in Edit mode

This design is an initial state of the application intended; the design is aiming for compelling ways to be *seductively* by the look and feel of the interface creating a playful novel concept that change dynamically while content is created triggering events and actions pushing the users to be more active in the act of share and spread the word. Collaboration is also a point treated in this design where users from different devices can access and work together creating content. Thus, every single detail treated in the meeting is saved hierarchically creating order and preserving the structure of the meeting to further revisions or to look up for answers or task still pending.

7. Discussion

This has been a remarkable project with great challenges, where it was raised a topic that was originally touched in the business/marketing domain to be translated in the interaction design domain.

In this thesis project I have carried with a large literature search, looking for possible approaches where other authors have worked with the concept of virality as a design driver. What I have found, is that most of the content had a strong relationship with marketing and business making very difficult to relate and find valuable content from an interaction design perspective. In that sense, what I did in this project was to look for concepts related with virality in order to establish a common ground to create a database of possible patterns, which in turn will be used as the theoretical foundation of the whole thesis.

Looking for Virality

This approach mentioned above, was really demanding because it required a lot of time reading many useless texts that work more as a marketing manual to apply selling technic to products already designed, instead of inspiration for design and creation of new products. However, something interesting that I found really valuable was that, as soon as is found a good text the references of this tend to be good too; Letting me dig more into details and information valuable for my research. It is like touching the top of an iceberg but then you realize that there is a huge mountain of information deep into the water. So, I found many valuable books and articles like "Made to stick" of the Heath Brothers (2007), this was an incredible book. Because, although, it is a marketing book with a very design focused approach quite convenient to start exploring into this concept of virality.

Then, authors like Seth Godin (2000), Christina Wodtke (2009), Adam Pennenberg (Pennenberg, 2009) and BJ Fogg (2003) help me a lot to create an understanding of the relationship developed between customers and products and how this relation could be used to enable elements in the design to provoke certain actions with the aim of spread the word rapidly in others to drive virality. It is very interesting how extensive this topic it is. Actually, I think this project is only an initial state of the universe of possibilities available for interaction designers. For example, Nostalgia is a component discussed in the viral pattern *Seductively;* this component only is an activator of many actions ranging from the purchase of a product to the development of an entire culture around the cult of a nostalgic element, as is the case of video game

nostalgia²¹ which explore the idea to design based in pixel art and composed music based on "chip-tones" like the old days games.

The Pattern's approach

After this extensive literature search, a good way to arrange all this information and make it reusable for me and the possible stakeholders was to use a pattern-based design approach, this approach let me document in a very neat and simple way how virality and collaboration could be expressed in design terms and how could be reused when it was necessary. This was a good way to arrange all the information. Because, otherwise I would have ended with an extensive document that describe the design done and readers would have had to find a way to interpret it and use it in their designs. I find patterns a good learning and solving problems solution. As I mentioned above in this document pattern by themselves do not work alone, they work connected similar like "Lego bricks" creating connection and solving complex problems. But, what happened if the designer is looking for a new paradigm and those "Lego bricks" are not enough? I think that have a spectrum boundary that from time to time should be break in order to come with new ideas, *unexpected* solutions. Additionally, it was hard for me try to arrange a hierarchical value of the patterns. Thus, I have to rely on the MDA model (Hunicke, et al., 2004) to order and rank the patterns in a structured way to define guidelines to design for virality and collaboration. What I mean with this is that Patterns in my consideration, requires a third element to connect them and create order on the way they are going to be used.

The Methods

The other methods used in this project, iterative design, prototyping and target users were a valuable tool to fulfill an reach the design level desired, iterating in the design is the only way is possible to improve an refine the quality of the design. As interaction designers, we have to deal with many inputs and outputs that require attention and are quite likely that sometimes those elements could be missed over time. That is the value of an iterative design and are find issues that otherwise it would be impossible to find. I consider that a collaborative approach through workshops in the iterative process is the key of a good design, and is always important encouraging new students to work in this way.

About the results

Now, discussing about the results of the thesis project; I am very pleased with the research done in the viral and collaborative patterns, this was a very interesting project and definitely I would like to continue working with this research. As I said before, the possibilities of future work related with this could open a new door for the

²¹ Why we get nostalgic about old games: <u>http://www.psychologyofgames.com/2010/11/why-we-get-nostalgic-about-good-old-games/</u>

creation a really compelling product that become really popular and will be spreader rapidly among the users. I can imagine projects done like "Static!" (Interactive Institute, 2006) Combined with the concepts introduced in this thesis, to go to the next level where is deploy the Static project in houses around Sweden. If this is carried correctly it could be a total success of products that help people to be aware of their energy consumption while they enjoy do it that and encouraging friend and family to adopt it. Converting this project in a viral success for the Swedish Energy Agency and the researchers involve on it.

On the other hand, something is missing is an evaluation of this patterns. But the problem of the evaluation is related with their temporal nature. Viral patterns can be evaluated over time, when they are released among the users. This topic is discussed by Pennenberg (2009), Which propose a formula called the "viral coefficient", where the product is released in the market and after a period of time is calculated the difference between the numbers of people expected to be affected and the number of people affected in that timeframe; If a percentage of people are affected in certain amount of time is possible to say that a product is viral or not. However, Emilio Gimenez (Student of Interaction Design at Chalmers University) is using this research, in his thesis project as the theoretical foundation for the creation of a system for Volvo cars to spread the brand beyond the car experience. So, maybe in a future when he is presenting his thesis, we can return to this project and evaluate the design patterns proposed.

In the case of the design I feel very confident that it has been done a good work in the aesthetical aspect of the application. But, I think is required to create a functional prototype in the future in order to test better the functions that are the mechanisms, which drive virality and collaboration. Those are the backbone of the dynamics proposed in the different patterns mentioned. So, this prototype is necessary to evaluate and improve those mechanics. This could be something to think in the future if the stakeholders decide to continue with this project.

Future Considerations

Finally, I consider that the future of this project relies in the evaluation and the connection of the different patterns creating relationships and hierarchies that combined with the MDA model (2004), create a solid structure to be released as a design tool solution. Moreover, the collection of patterns could be extended and increased, I propose use this document as an initial stage of a bigger research where many other can contribute looking for patterns and increase the content, creating a powerful tool for design and push the idea of viral designs.

Since this project, offers as a result the viral/collaborative patterns and the application design. Improvements should be carried in both areas. In the case of the patterns relations and the addition of more patterns should be included. Even more, the inclusion of anti-patterns would be a really valuable contribution since create a manual to define boundaries in the designs proposed. So, the patterns could find a

limit when they clash with the anti-patterns creating an interesting complex amalgam of concepts that will help designers in the creation of their viral designs.

In the case of the design, implementation, testing and evaluation is necessary in order to validate the previous research answering many unsolved questions open in this research.

8. Conclusions

As a conclusion, this thesis answers the research question, "How can we, in design, utilize virality as a means to drive a computer-based collaboration among the users?" with a collection of viral and collaborative patterns that can be used as a design tool along with a set of design guidelines. The thesis also exemplifies how the patterns are being used in the design of a viral collaborative tool. Arguably, the design has not been tested yet, since this requires a fully working product and a long-term study. However the patterns are still validated in the sense that the collection has been harvested from existing literature.

Virality understood as a set of patterns, works as a powerful driver for design, providing valuable instruments for interaction designers. Designers that choose to work with these patterns will find how business and marketing strategies have been turned into an interaction design approach, where it is possible to design for virality, rather than (only) using marketing and PR to create this effect.

The hierarchy of patterns and the additional design guidelines presented in this document have solved the research question. Therefore, we can, in design, utilize virality as a means to drive a computer-based collaboration among users through the use of patterns applied to the design, taking into account that patterns do not work as fragmented pieces, but rather as a connected whole.

9. References

Alexander, C., Ishikawa, S. & Silverstein, M., 1977. *A Pattern Language: Towns, Buildings, Construction*. New York(NY): Oxford University Press.

Allen, D., 2001. *Getting Things Done: The Art of Stress-Free Productivity*. New York(NY): Penguin Books Inc.

Anderson, A., 2012. *Why We Share - Interactions'12*. [Online] Available at: <u>http://www.ixda.org/resources/angel-anderson-why-we-share</u> [Accessed 09 11 2012].

Anderson, S. P., 2011. Seductive Interaction Design: Creating Playful, Fun, and *Effective User Experinces*. Berkley(CA): New Riders.

Bergström, K., Björk, S. & Lundgren, S., 2010. *Exploring aesthetical gameplay design patterns: camaraderie in four games*. New York, ACM.

Bjork, S. & Holopainen, J., 2005. *Patterns in Game Design*. Hingham(Massachusetts): Charles River Media Inc.

Borchers, J., 2001. *A Pattern Approach to Interaction Design*. NewYork(NY): John Wiley & Sons Ltd.

Chevalier, F., Auber, D. & Telea, A., 2007. *Structural analysis and visualization of* C++ *code evolution using syntax trees*. New York, IWPSE '07 - ACM.

Cooper, A., Reimann, R. & Cronin, D., 2007. *About Face 3 - The Essentials of Interaction Design*. Inidianapolis, USA: Wiley Publishing, Inc.

Crumlish, C. & Malone, E., 2009. *Designing Social Interfaces*. First Edition ed. Canada: O'Reilly Media, Inc.

de Moor, A., 2006. *Community Memory Activation with Collaboration Patterns*. Prato, Italy, CIRN.

Dunham, R., Olson, M., Johnson, B. & Weaver, G., 1986. Using a Computer-Based tool to Support Collaboration: A field Experiment. Austin, Texas, MCC Conference, pp. 343 - 352.

Fogg, B., 2003. *Persuasive Technology - Using Computers to change what we think and do.* San Francisco, USA: Morgan Kaufmann.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J., 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. New York(NY): Addison-Wesley. Gladwell, M., 2002. *The tippng Point: How little things can make a big difference*. New York(NY): Oxford University Press.

Godin, S., 2000. Unleashing Ideavirus. New York, USA: Do You Zoom, Inc.

Goldenberg, J., Libai, B. & Muller, E., 2001. Talk of the Network: A complex systems look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, February, 12(3), pp. 211 - 223.

Hart, C., 1998. *Doing a Literature Review: Releasing the socialScience Research Imagination*. London: SAGE Publications Ltd.

Heath, C. & Heath, D., 2007. *Made to Stick - Why some ideas survive and Others die.* New York, USA: Random House Publishing Group.

Hunicke, R., LeBlanc, M. & Zubek, R., 2004. *MDA: A Formal Approach to Game Design and Game Research*. San Jose, California, s.n.

Iacob, C., 2011. Identifying, Relating, and Evaluating Design Patterns for the Design of Software for Synchronous Collaboration. Pisa, Italy, ACM, pp. 323 - 326.

Interactive Institute, 2006. [Online] Available at: <u>http://dru.tii.se/static/</u> [Accessed 30 May 2013].

Jacobson, I., 1992. Object-Oriented Software Engineering. s.l.:Addison Wesley.

Jenkinson, A., 1994. Beyond Segmentation. *Journal of targeting, Measurement and Analysis for Marketing*, 05 31, Volume 3, pp. 60 - 72.

Kollock, P. & Smith, M., 1999. Communities in Cyberspace. London, UK: Routledge.

Leskovec, J., Adamic, I. & Huberman, B., 2007. The Dynamics of Viral Marketing. *ACM Transactions on the Web*, 1(1), pp. 2-39.

Lundgren, S. & Björk, S., 2012. *Neither Playing nor Gaming: Pottering in Games.* s.l., ACM.

Mantei, M. & Torey, T., 1988. Cost/benefit analysis for incorporating human factors in the software lifecycle. *Communications of the ACM*, April, 31(4), pp. 428 - 439.

Nielsen, J., 1993. Iterative User Interface Design. *IEEE Computer*, 26(11), pp. 32 -41. Pennenberg, A., 2009. *Viral loop: The Power of pass-it on*. London, UK: Sceptre.

Ries, A., Trout, J. & Kotler, P., 2000. *Positioning: the battle for your mind.* s.l.:Mc Graw Hill Inc..

Robinson, D. & Reed, V., 1998. *The A-Z of Social Research Jargon*. Cambridge: Sheffield University.

Rodriguez, F., 2012. *Viral Design - Understanding Virality from a persuasive Perspective*, Malmö, Sweden: Malmö Högskola.

Schön, D. A., 1983. *The Reflective Practitioner: How Professionals Think in Action*.. New York: Basic Books - Perseus books Group.

Schummer, T., 2004. *Patterns for Building Communities in Collaborative Systems*. [Online]

Available at: <u>http://www.pi6.fernuni-hagen.de/publ/SchuemmerEuroPLoP04.pdf</u> Smith, M., 1991. *Software Prototyping: Adoption, Practice, and Management.* London: McGraw-Hill.

Stewart, M., 2008. *Wikipatterns, A practical guide to improve productivity and Collaboration in your organization*. Indianapolis: Wiley Publishing, Inc.

Tidwell, J., 2005. *Designing Interfaces*. First Edition ed. Sebastopol, CA: O'Reilly. Whittaker, Z., 2009. *10 technological changes in 10 technological years*. [Online] Available at: <u>http://www.zdnet.com/blog/igeneration/10-technological-changes-in-10-technological-years/3274</u>

[Accessed 23 April 2013]. Wodtke, C., 2009. *Designing the Viral App*. [Online] Available at: <u>http://www.ixda.org/resources/christina-wodtke-designing-viral-app</u> [Accessed 30 10 2012].

10. Appendices

Appendix A

Project Schedule





Appendix B

Collaborative Patterns

Collaboration patterns are a particular class of patterns that capture socio-technical lessons learnt in optimizing the effectiveness and efficiency of collaboration processes (de Moor, 2006).

1.1.1. Goal Patterns

These kinds of patterns are related with the conceptual representation of group and individual objectives (de Moor, 2006). For instance, the group objective could be the creation of a new product and the individual objective will be the planning and the creation of the different parts of that product such as; design, engineering, etc.

Annotate

Annotated is in itself a thread-like entity that allows collaborators to request back thoughts or important ideas through text, audio or video (Iacob, 2011).

Example 1: A good example of this pattern is what happened in Google Slides²² where collaborator can prepare a keynote and in the bottom of the presentation they have the ability to add comments related with the presentation itself. These comments could be used as a help to the presenter and also as a helper to follow the ideas of the collaborator.

Using the Pattern: Annotate is an important part of a collaborative application, because let other collaborator to understand the ideas, concepts and decisions taken during the creation of a specific content. This pattern requires a special treatment because the abuse of it could create a mess of data in the application generation an excess of information. Usually this could be implemented as a parallel function that is attached to the data of the collaborative application.

Structured View

This refers to the possibility that people have to look the entire information through a structured view having a holistic view of the document over time.

Example 1: PiratePad²³ is a novel concept that let users to create text based in the collaboration of several people. In this application there is a function called "Timeline Slider" where the users invited or anyone who has access to

 ²² Google Slides: <u>http://www.google.com/intl/en/drive/start/apps.html#product=slides</u>
 ²³ PiratePad: <u>http://piratepad.net/iTEPHKcJko</u>
the application can see the content created over time, enabling collaborator to understand how the content has evolved and who create what.

Example 2: An structured view could be represented in several ways, For example if a text is written starting with a keynote and then people starts commenting and adding content on it based on those topics, developing a structured view which is possible to see over time understanding who starts the discussion and how this one has evolved. This is the case of Rizzoma²⁴ a collaborative discussion board system born after the termination of Google Wave²⁵; that let user structure the content based in the development of this. It is possible to perceive how the content is done through a mind map structure or an accordion view that show/hide the content created.

Using the Pattern: Usually in a collaborative environment is created lots of information which require a neat structure to understand easily who, when and what has been done. In that way, when the designer applies this pattern is required to think in the quantity of information and types of information (Ex: Text, Audio, Video, Etc) used. A bad structure could produce an inconsistent configuration difficult to understand, avoiding simplicity to make it viral and usable.

Collective Intelligence

This pattern occurs when some problem is looked under collective intelligence paradigm the focus is on connecting the intelligence of the group to enhance productivity and enabling better decisions to solve the issue; different from the approach of those individuals working alone to find a solution.

Peering

This pattern happens when two or more user voluntary agree to connect in order to exchange information and solve certain issue that is attached to both parts.

Folksonomy (Tagging)

This refers to a classification system based in the creation and assignation of tags that categorize the content done by the same collaborators. This pattern allows collaborators collectively to classify and simplify the way the information will be found and indexed (Stewart, 2008).

Example 1: There are plenty of examples in the latest web application available nowadays. For instance, del.icio.us a collaborative bookmarking

²⁴ Rizzoma – Collaborative Board : <u>http://rizzoma.com</u>

²⁵ Google Wave Status: <u>http://support.google.com/bin/answer.py?hl=en&answer=1083134&rd=1</u>

website where people add their favorites website to then, other users look for terms finding bookmarks tagged based in the people's criteria. In this way the content is cataloged by a general consensus and give priority to the most relevant bookmarks.

Example 2: 43 things²⁶ a collaborative inspiration website that make use of tagging to inspire people to share their goals or make use of others goals to get inspire is an interesting example of how folksonomy provide an environment where people's contributions help other people.

Using the Pattern: Design for tagging is a matter of definition which kinds of elements are more important to classify and identify in order to create and structure based on social tags. Tags are valuable, but not necessary in every single piece of content. For example tags can be used to group main documents making them easy to find for new and old users.

1.1.2. Communication Patterns

They are collections of related communicative workflows and rules definition, describing suitable communicative interactions with the group. For example:

Socialize

This pattern addresses the mechanism that let users interact with each other. This pattern could be presented in several ways but in general is the way users look other, communicate with others and perform task along other people.

Acknowledge Goodness (Recognition)

This pattern indicates the use of tools that help to recognize the value added to the content by the contributors (Stewart, 2008). Usually, this pattern is driven by the personal motivation of the collaborator to be acknowledged and respected by his/her peers.

Example 1: Yammer, the social corporate network help to recognize the people that collaborate and interact more with the network through automatic post and banner that shows the recent activity and the content created by other, encouraging people to visit and inviting them to be part of it.

²⁶ Collaborative folksonomy: <u>http://www.43things.com/</u>

Example 2: Another example Acknowledge goodness is the BarnStars²⁷ granted to the people who collaborate more and make the most valuable contributions of the Wikipedia. This award has become a standard in different wikis and basically shows the hard works of their contributor. This is a recognition is placed in the bio page of the contributor, allowing others to check the work and reliability of the contributor.

Using the Pattern: When is designed for acknowledges goodness is important to notice that this pattern does not offers direct impact in the content. Instead, this has a positive impact in the collaborative culture generating an environment of fellowship and collaboration. Additionally is important to avoid the over-use of it, a delicate balance granting the right recognition is required.

Kick Off

This pattern happens when it is initialized for first time an application, introducing the collaborator to the basic functionalities of the application and setting up an introduction to begin the work.

Example 1: "Getting started with Notism is really easy..." that is the message display by Notism²⁸ a collaborative application to discuss visual content. Here the application offers three steps to follow in order to get started into the application. Basically, the user is instructed about the application's workflow, how to invite people and an interactive tutorial that conduct the user to his/her very first project.

Example 2: Another example of application's kickoff is the one provided by Pipedrive²⁹ an application to manage sells and customers in a very simplistic way. Based in a pipe paradigm where each pipe has to be clear to move to the next one. The kickoff of this application starts with a popup windows showing a video that explains the basic functionalities of the application and the key features to start using it.

Using the Pattern: The use of this pattern is widespread today. Simplicity is the key of success when is designed the kickoff of the application.

Supporting Communication

This pattern addresses the problem of supporting communication among

²⁷ Wikipedia contributors recognition: <u>http://en.wikipedia.org/wiki/Wikipedia:Barnstars</u>

²⁸ Notism: <u>https://www.notismapp.com/tour</u>

²⁹ Pipedrive: <u>http://www.pipedrive.com/en-US/features</u>

collaborators. This support could be presented in several ways like commenting systems, chats or others. This pattern allow distant collaborator to communicate to avoid misunderstanding, clarify ideas and/or express concept to the team.

Example 1: The whole Google-docs platform is an example of communication; the platform offers a chat system along the applications, which differentiate the chatters by colors and users-name. Additionally, is possible for the user know who is online and what was the last time somebody communicate something to the team.

Example 2: Collabedit³⁰ is a collaborative text application that let people write text in several programming languages. Here contributors have a simple chat that identify the collaborator by their name sorted in descendent order from the oldest to the latest message posted in the chat.

Using the Pattern: This patter is present in many ways email, chats, comments, etc. But design for this pattern has some flaws; first there is a thin line between good communication among the participants and the excessive flow of messages creating spam among the team. So, is important to measure the impact of every single communication mean provided to the user. If is totally necessary provide the user with a communication tools if not, it is better try to cover that need with software functionalities. For example, instead to have a chat to ask how the system works the system should provide a better and usable interface or a help system that teach the user how to interact with the application. Novel communication means can be tested. It is not necessary to be attached to the traditional ones.

1.1.3. Information Patterns

The information patterns are associated with the content generated and the responsible actors. For example:

Structuring Content

This pattern addresses the mechanisms to hierarchize content allowing collaborators to determine what is important and most valuable into the content.

Example 1: Is common that application for project management and GTD make use priorities to determine what is more important to do in a define schedule. For instance, in Asana an application previously discussed, the user can define what will be important and more relevant to get done.

³⁰ Collabedit Collaborative Text: <u>http://collabedit.com/a9vxq</u>

Using the Pattern: When is designed for this pattern, in valuable to understand what type of content is going to be hierarchized. For example, if the content are groups of mind-maps developed in brainstorm sessions the treatment that the users can give to this is quite different that documents for presentation to the direction board. In the first case the user criteria is the most important since are concepts that has to be polished. But, in the second case the hierarchy is determined by certain demands done by the board. In that case is important to validate this kind of concept an evaluate the need or not of this pattern.

Privacy (With or without collaboration)

This refers to the capacity to provide a private area not available to the other collaborators, where the owner has complete control of the content (Iacob, 2011). This pattern usually is use as a temporal space to testing, writing thoughts or prototype ideas.

Example 2: Salesforce³¹ let user determine the privacy level of the documents uploaded to the system. For example, if someone has uploaded a document, this one starts private and then the user can decide if he/she wants to share it with his/her coworkers or people outside the company!

Using the Pattern: When is designed for privacy it is important to understand the purpose of sharing on it and how sensible if the information that is used in the collaborative application. Having that into account,

Previews (Eye wide open)

This pattern addresses the possibility that have other collaborator to visualize what he/she and others has done during the contribution process. Allowing each collaborator to be notified about and visualize what the others are contributing to the process at any time.

Example 1: PiratePad shows a highlighted look of what the different participants of the text has written. The text is differentiated adding an specific color tone letting viewers or newcomers understand the contributions done by the writers.

Using the Pattern: Designing for this pattern let new users have a better understanding of the work done during the collaborative process. This pattern requires the use of different information visualization techniques to compile lots of

³¹ Salesforce CRM: <u>http://www.salesforce.com</u>

information in an understandable way. With that is possible to embrace a wider group of collaborator, which in turn is going to result in a better adoption of the tool.

Helping

This pattern is a key mechanism to success in a collaborative environment; showing an "incremental path" of the different changes done in the content is a way to help users to understand what is happening and how the content has evolved (). Additionally, this pattern offers a broad concept where helping tools are key to accompany the user during the collaborating journey offering to him/her support and a knowledge base to answer questions.

Example 1: Google docs provides several helping channels to their users, one interesting tool that they provide is a modal preview that comes after a key combination (command + / for mac users) this window shows all the keyboard shortcuts designed for those power users that prefers keyboard commands instead of slowly menus commands. Summed to this user have simple helpers like "Who's online" status bar...

Example 2: Asana provides users with a quick view of the most common used keyboard shortcuts in the bottom of the interface. Additionally they provide a link to a help center, a knowledge database filled with frequent question and tutorials.

Using the Pattern: Detailed pieces of valuable information in the right moment are the success key for helping. A user facing complex applications requires assistance to the different tasks they have to done.

Indexing

This pattern is based in the grouping and organization of the content created by the collaborators facilitating their search and location. This pattern is definitely dependent of "tagging" performed by the collaborators ().

Example 1: Evernote³² let people add tags to notes in order to group them in different categories. Then these notes can be shared and populated with content by others. Letting them to add their own tagging, which creates an interesting way to categorize, based on people criteria to simplify searches and organization in one's mental map.

Example 2: As is mentioned before del.icio.us has a complete tagging system to create social bookmarking permitting people around the word share and

³² Evernote taking note app: <u>http://evernote.com/</u>

collaborate with others tagging, categorizing and complementing their favorite websites. This content is indexed in a huge database that can be public simplifying the search of a specific site thanks to the tags and categorization done by others.

Using the Pattern: This pattern could be used to bring order and simplify a search for the user. Definitely this pattern should leverage on others as tagging, structured and hierarchy.

Structured

This pattern is present when collaborators start an application with a fixed content that it is editable. In other words is the capacity to a template or framework to start working instead to create everything from the scratch in an empty canvas (Stewart, 2008). This pattern let people to have an organizational concept of the content and the element needed to fulfill some purpose or goal.

Example 1: Wikipedia offers to its contributors, templates³³ to create fast and easily content. The concept behind of this is allow user to concentrate the writers more in the content instead of the style of it. Simplify a lot the work performed, encouraging them to continue adding more content to enhance the whole encyclopedia.

Example 2: Slidifier³⁴ is a simple presentation tool that has a fixed theme and a structured manner to create content that permit people create content easily to be presented to others.

Using the Pattern: Design for this pattern is directly related with the core of the application. If the application is a task-management tool, the structure should be predesigned thinking in the most common uses for a task, like time, title, date and assigned people. The key to design for structured is to be aware to cover the needs of the most and not the needs of just a few ones.

1.1.4. Task Patterns

These patterns describe the Who/What/When and how, the content created by the information a communications patterns are going to be used.

³³ Wikipedia Templates: <u>http://en.wikipedia.org/wiki/Wikipedia:TEMPLATE</u>

³⁴ Slidifier : <u>http://slidifier.com/slidifier.html?id=ywk58h2SKv&key=Tg3ApXcdfq</u>

Scheduling

One person cannot perform alone all the tasks; this means that other collaborators have to participate in order to contribute building the content together. However, people have different tasks and errands to do in their jobs with no-matching agendas (Stewart, 2008). This causes the need of an agreement system to define a common agenda for the collaborators inside a project.

Example 1: Wikipedia have an agenda that let collaborator receive emails to know when is the next session to arrange or add content in a wikipage. This allows people to know what others have done and when is the next working time.

Using the Pattern: Designing for scheduling is an easy one to treat basically what is important here is making easy for the user understand the schedule and how this can affect others. For that there are several approaches already discussed in "About Face 3" (Cooper, et al., 2007), which could be used as a reference for this case.

Assigning

Since collaboration is not only a matter of agreement of a common agenda but it is a matter of responsibility. So, this pattern refers to the need of assignation of tasks and duties into a project development. Which in combination of scheduling people can define times and deadlines for specific tasks.

Example 1: When a manager requires to some of his/her sellers assist or visit a costumer, he/she can assign that meeting to them. The application allows managers to check their sellers' agenda. Which makes easier arrange and fix time among the working team.

Example 2: Microsoft Outlook is one of the most common collaborative applications that have this kind of pattern present. For example when a person looks in a workmate calendar in outlook and he/she can assign a task to do or invite this person to be part of a meeting. Assigning time into someone else schedule.

Using the Pattern: Assigning is quite related scheduling; here the user is assign to perform some task. In that way when is designed for assigning is important to understand the communication channels and design elements that clarify and simplify the way this person accept his/her assignations.

Hyperlinking

In a collaborative environment could has a lot of content generated by several people, creating an intricate network of information that follows a specific thread. This multidimensional relationship requires mechanisms that let the users know who did what and who is working on what.

Using the Pattern: the designer has to understand where the users are interacting and apply helpers that simplify the understanding of the networked information. For that reason the designer not only create the elements but also has to take into account the interaction performed constantly by the user which can change constantly depending of the action performed by this. In other words, is important to provide clues to the other user with the aim of create a guide to understand what the present users in the application have done.

Team-Up

This pattern appears when different teams want to share content each other. A team has developed a content related with a project and they need to share that information to other stakeholders or teams (Stewart, 2008). So, this pattern refers to the mechanism that allows sharing and growing the content not only in the core team but also in other team in an organization.

Example 1: Files can be shared with anyone in Dropbox, even the files that were shared by others. This is an example of how when a team is established some else in the team can invite other teams or individuals to work together with a specific project.

Using the Pattern: In this pattern is important to take into account if the application is designed to work as a stand-alone or for multiple teams. Design for team-up involves the creation of tools to create teams, arrange agendas and then share content with the other teammates. The use of this pattern is valuable for large design teams' student groups that sharing parts of a joint project with a third party.

1.1.5. Meta Patterns

These patterns are related with the quality of the other patterns and how is ensure those are good. For example:

Versioning, Changes and Evolution

This pattern shows the evolution of the content developed by the collaborators. This mechanism allows having several versions of the content permitting users modify and interact with the content on different stages.

Example 1: Dropbox keep several versions of the same file in different stages. With this is one of the collaborator by mistake destroy, modify or make a mistake is very easy to fix it returning to a previous version. Additionally this functionality makes possible for other user know how many changes a document has had.

Example 2: As soon as one of Wikipedia's contributors makes a change into the content, this one makes a copy of the previous article in order to back up the information. With this, Wikipedia can keep track of the different changes and in case of destruction or abuse the can return to the previous version.

Using the Pattern: When is designed for versioning the designer has to think in the users as hierarchies. Where some user can see the evolution of the document but other with a higher hierarchy can modify or restate a previous version. However, in any case the need of a neat interface where makes clear the evolution time line makes a great contribution to the simplicity of the application.

Consensus

This pattern refers to the step taken by a group to agree in a topic discussed during the collaborative session. Decisions are taken by a general agreement among the participants and usually conducted by the project leader.

Example 1: This more than a mechanism or functionality in a computer-based collaboration, works as arrangement among the participants of the collaborative event. Usually, the use of chats or other means like voice or video are enough to determine an agreement and decide if the work has been done. For example, in Google docs people have a chat in the right side of the interface this is used as a communication tool to define topics and clarify doubts. When the document, Presentation or spreadsheet is finished a leader ask for a final review to check close the document to send it or simply save it.

Using the Pattern: When is designing for this pattern the designer has to face the difficulty of imagine a mechanism that helps the users agree in some issue. Usually this is fixed by a simple question in the real world. However, in computer based collaboration some means like chats can supply that channel. Here the designer can explore clever ideas to create new tools that bring consensus in the decisions taken. Normally, this requires several iterations until a useful system comes afloat.

Sessions

This pattern occurs when the collaborative process is interrupted by some external factor like time or space constraints. In that way the collaborative sessions is interrupted creating a "Stand By" state to be re-taken later by the collaborators.

Example 1: When

a person leaves the Twitter client for iOS and then return the user starts again from his latest tweet but is advice of the newest tweets available. That let the user check all the messages written since his last time in the application.

Using the Pattern: Session basically requires a status of what was the latest state of the user in the application. However, in a collaborative environment this could be tricky since other could modify the latest state. So, the versioning pattern can work as a helper to let the user understand his/her previous session plus the addition or deletion done by the other user. Helpers and simple indicators are necessary to show this evolution in the session map.