ABSTRACT
Ever since the release of the first graphical user-interface, user-interfaces of the modern age are full of metaphors. They help us get a better grasp of the user-interface, how we can interact with it and what it can help us achieve. With the rising of ubiquitous computing and each user owning more than one devices, it is crucial to design in a fashion that is easy to understand even for people with less knowledge about technology. Metaphors exploits the person’s prior knowledge of a source domain in order to use it into a different target domain. In user-interfaces, designers use the user’s prior knowledge in order to make the interface easier to understand and more enjoyable. This paper analyses literature behind user-interface metaphors. Aim of this research is to find out how metaphors help users understand the user-interface, and how to use metaphors efficiently in designing interfaces. The paper investigates the common terms that relate to interface metaphors and how metaphors from other fields relate to it. Some tips and suggestions from papers on metaphors have been collected on how to design with metaphors. Questions regarding the usage of metaphors in interface design and future research have been asked.

Keywords
User-Interface, Metaphors, Affordance, Metaphoric Entailments

1. INTRODUCTION
In modern User-Interface design, designers try to create interfaces that are user-friendly and try to cause as less frustration and anger to the user. They tend to design for usability and portrait the interface in a way familiar to the user, so that it is instantly recognisable and easy to operate. They try to bring into their interfaces familiar knowledge to the user, even from the real world. Sometimes, they design metaphorically. Even since the first Graphical User-Interface(GUI), metaphors is one of the most commonly used approached on designing interfaces. The first GUIs Xerox Star/Alto and Apple Lisa/Macintosh, were designed around a metaphor, that continues to exist now, nearly thirty years after (Figure 1). The first GUI was designed around the desktop metaphor. The user could stack pieces of papers around his desktop and drag them around and place them in folders, like he was used to in the real office. They were designed and marketed as a powerful tool for the workspace. It would only make sense if this powerful machine that was promised to replace all the paper work of the office space, would resemble the previous tools people were used to use. The problem is it looked and behaved anything like a desktop. It was the meaning that the user made out of all this virtual representation of the desktop that made sense[22].

2. METAPHORS
Metaphors are still a commonly used tool in designing modern interfaces. Personal computers are still being designed around the desktop metaphor, e-shops contain shopping carts in order to carry your virtual considerations, applications that look like magazines or books that are able to be flicked through by turning their pages are just a few to name. Metaphors have their place in the academic and commercial fields. The "big" companies such as Apple, Microsoft and Google not only encourage developers and designers to use metaphors in their designs, but they also tend to design sometimes around them. According to Apple’s OS X Human Interface Guidelines "a good way to take advantage of people’s knowledge of the world is to use Metaphors to convey objects and actions into the app)[6]. Google’s latest design
principles, Material Design, are based around the metaphor that pixels are made out of a magical material, that behaves as something that exists in the real world. We are quickly coming to an era that each person has more than one computational device with them. Computers are now being used everyday by people that are not necessarily technology enthusiasts. People that are considered to be less 'tech-savvy' tend to struggle using all these new technologies. Metaphors are a great way of introducing new and novice users into new pieces of technology. They are used in order to portray a system functionality the user has never come across before, by using similarities and ideas from the real world the user is familiar with[1]. Using the existing knowledge about a concept, the user can easily figure out how the software works and master it in no time. The structure of this paper is as follows: First, a first look on metaphors is going to be given and related work on metaphors is going to be mentioned. Secondly, the work of Lakoff and Johnson is mentioned as a key ingredient for research on metaphors and how researchers have extended their work in the field of user-interface. Common terms such as Affordances and Metaphorical Entailment and their connection to metaphors will be given in a following section. The last section contains a list of tips found in various papers regarding on how to design with metaphors. Lastly further questions and future research can be found in the discussion of the paper.

3. RESEARCH ON METAPHORS

The user-interface metaphor is as old as the graphical-user interface. The concept of metaphors is taken from two different fields, linguistics and philosophy[10]. A considerable amount of research has been done on metaphors. Despite that there seems to be no distinct definition of what a user-interface metaphor is. Most definitions are taken from the other fields. Metaphors are a essential part of our everyday life. We talk and think in metaphors, and everything we do is a matter of a metaphor[11]. We tend to think and talk in metaphors without even realising it. Our learning is so tightly related to creating relations to past experiences. Metaphors are known to have the explanatory powers[14]. Because of this ease of introducing people to new complex ideas, metaphors are being used in the design of user-interfaces and technologies. One way or another every user-interface contain some form of metaphor. The reason why metaphors help the users use an interface they have never come across before, is because of their own previous existing knowledge. They do not reduce complexity; they increase familiarity[9]. A metaphor in User-Interface design is not necessarily visual. It can consist of other senses as well such as hearing (sounds) or multiple images[1]. Metaphors, by definition, provide inexact mappings between a source and a target object[8, p. 67]. Through the literature, user-interface metaphors can be seen as a way of explaining complicated system functions to the user, exploiting their prior existing knowledge, taken from different domains and/or user-interfaces. A metaphor can be a powerful tool. It helps us understand new concepts[19]. Designers use metaphors not only in their designs. They use metaphors in order to brainstorm new ideas[4] and use techniques based on metaphors in order to design[7]. If a system behaves similar to some other system the user has used before, the learning time is greatly reduced[21]. That is the reason why metaphors create this form of familiarity with the product. With a quick look at the interface, they already know how to use it, just because of their familiarity with the source object of the underlying metaphor. The use of metaphors can often expose to the user what the object behave and how it can be used. The target object of the metaphor often keeps many of the source object’s attributes and characteristics. By using metaphors, the user, regardless of computer experience level, already knows the majority of the interface behaviour. The interaction between user and device is simplified. Thus, little to non explanation is needed[21] and the user can use the interface faster. That means that metaphors work as a navigational aid to the user. By designing the metaphor in a way that is expected to the user, it can produce iterations that are natural and intuitive to the user[12].

Despite the advantages of the metaphor in user-interface design, metaphors have been criticised through literature. A metaphor is a great way to introduce new system functionality and reduce the complexity of an interface. As long as the user has mastered the interface, and what the metaphor is being used to describe, the metaphor itself is becoming a "dead" weight[16]. Novice users already know how to navigate through the interface, and advanced users do not need a metaphor in order to understand the interface on the first place. According to Nelson[16], metaphors cause the user to learn software by "approximation rather than understanding". By breaking the metaphor though "magical things can happen"[21]. Without having the constrains of the metaphor in mind, we can make full use of our computational powers and start thinking in new concepts that are not yet been thought of. Designing with metaphors can be really hard as well. It might be easy to find a visual metaphors for objects, such as a printer, but thinking for a metaphor for some service, system or function might be hard if not impossible[5]. Lastly, they limit our computation power. Take the clipboard for example in the desktop metaphor. It can hold only one item every time. Also there is no information on where the item came from. Metaphors give a great momentary boost to novice users, but at the cost of sacrifice system capabilities. Metaphors are also strongly connected to culture. There are many that criticise the desktop metaphor. When the first graphical user-interface was released to the public, it was marketed as a office replacement tool. The user could do all sorts of work on their computer the same way they would in the office space. The metaphor works only if the user has been in a office. What happens if that is not the case? Research shows that it is possible to replace the desktop icons to items with strong cultural influence, and people would still understand their meanings, regardless of being in an office[9]. Let us not forget the fact that the desktop metaphor was 'invented' almost thirty years ago. Hints of it can be still be seen in our devices that have moved far away from the office, or any room.

4. EVERYDAY METAPHORS

It is important to understand how metaphors in the real world operate, in order to be able to take full advantage of them in our designs. Lakoff and Johnson take a different approach on metaphors that we use in everyday life[11]. They set different types of metaphors into categories, making it easier to categorise them and use them. Different researchers have extended Lakoff and Johnson’s work in the field of HCI. In this section, the categories proposed by Lake-
off and Johnson are being discussed, and how they relate to user-interface design.

4.1 Structural Metaphors
Structural metaphors connect the existing knowledge of a structure to an other structure it tries to describe. Similarly in interface design, structural metaphors give to the simplified object the attributes of a structure out of the real world. A good example is the MacOS trashcan. The idea of the trashcan is already known to the user. The trashcan as a concept contains all the items are no longer wanted or needed, and are marked as unwanted/for deletion.

4.2 Orientational Metaphors
"Orientational metaphors give a concept a spatial orientation"[11, p. 39]. Orientation metaphors are connected to space. We use directions to convey meanings such as MORE IS UP; LESS IS DOWN, HAPPY IS UP, SAD IS DOWN or IMPORTANT IS UP. They are tightly connected to culture. According to [11] in some countries the future is pictured as forward, but in others as back. That is because, in left to right languages, the words and sentences are read from left to right. They start from left (past) and continue reading to the right (future). Because of their strong relation to our physical and cultural experiences, they do not appear to be metaphorical[2]. User-Interfaces are full of orientational metaphors. The same concept of MORE IS UP; LESS IS DOWN can be seen, for example, in music players. When the volume bar rises, the music is louder, but when it lowers, the volume goes down as well. An other example would be the passage of time in social networks. Newer "posts" and "status updates" are displayed on top of the "newsfeed"; older ("past") "posts" appear lower in the "newsfeed".

4.3 Ontological Metaphors
Ontological metaphors give a system concept or some method its own entity. In that way the function itself can communicate with the user. In interface-design, ontological metaphors is also a common type. They help the user connect some functionality with some existence of the real world. A computer process for example can have their own behaviours and needs (ERROR IS ENTITY)[3]. For example, when the user tries to perform some action an error might occur. The error "requests" the user attention, "prevents" the execution of the program. In the operating system iOS for example, "applications request for permission" in order to perform some action. Applications are portrayed as entities (Figure 2).

4.4 Conventional Metaphors
Conventional metaphors exist in user-interface. Users are familiar with conventional metaphors because they have already come across their usage in other systems or applications[3].

4.5 Novel Metaphors
When working on some new User-Interface, a new metaphorical interface can emerge[3]. Those kind of metaphors are not yet familiar to the users. They have not yet found any similarity with any other known concept in any other related system. Such metaphor can be observed in MacOS X’s traffic light window control. The red light "stops" the window, terminating the program. The yellow light "pauses" the window, minimizing it into the dock. The green let’s the window "go", maximizing it to full screen. At first it is hard to spot the metaphor, or how it relates to the traffic lights indicators. It takes some time to see the metaphor and adjust to it. Novel metaphors can turn into conventional with the passage of time.

5. AFFORDANCES, CONSTRAINTS AND ENTAILMENTS
The term affordance is common to find in the literature regarding user-interface design and user-interface metaphors. All objects, real life and digital alike, have affordances. The relationship between the object and the person, and all the actions they can do with it are called affordances[17]. Affordances gives clues to the user on how to use it. For example, a door’s handle has the affordance of grabbing, and the door flat surface affords pushing. Perceived affordances also create expectations on how the object would behave[15]. For example, a chair has a steady surface. This can be used for sitting down, or placing objects on top of it and so on. Objects need to be designed in a way that they reveal their usage and way of using it by their looks. A good designed object can be used without instructions. Affordances are mentioned many times when talking about metaphors[23, 18, 15]. A user-interface metaphor transfers the attributes and characteristics of a real world object to the interface and that gives to the user some idea of how the object behaves and interacts.

Take digital forms for example[8]. Digital forms afford some of the same tasks, paper forms have. They both afford being filled in, one being by writing and the other by typing. In the paper form though, the user is allowed to fill in their details anywhere they like in the paper, and they have no constraints of where to write. Digital forms on the other hand, most of the times, have specific boxes that are need to be filled and, in fact, can only be filled. An other difference is that the abstract object can have more characteristics than the object that was referred from. A nice features some digital forms have is the ability to automatically validate the input they have received as soon as the user has entered data in them. Apparently, paper forms do not do this.

5.1 Metaphor Entailments
When using metaphors, characteristics and attributes are being transferred from the source domain to the target domain[2, 3]. This helps the user gain some understanding of the signified. For example, in most modern operating systems, the users can manage their files in the data storage.
The data storage itself implies the metaphor of filing. In the real world office, this can be done by using folders. In the desktop metaphors, files can be grouped together, placed inside folders. Like the real world, the folder can contain multiple files and images. Some metaphor entailments are not applicable though. Even though folders in the real world are being stored in drawers, there are not any to begin with in the desktop metaphor. The designer needs to analyse which metaphorical entailment is applicable to the current system.

6. DESIGNING INTERFACE METAPHORS
Metaphors are used in user-interfaces by designers all the time. Designers though do not seem to truly understand how to use them[1]. Despite the amount of work on user-interface metaphors, there is no standard way of using metaphors in designing interfaces. This makes it hard for designers to design proper metaphors. In this section ideas and suggestions on how to properly use metaphors are being accumulated for future research:

6.1 Balance prior knowledge with Power
Even though metaphors are a great way of introducing new technologies and ideas to novice users, it is important to utilise the abilities of the computer as well[6]. A mixture of real life concept with the magical capabilities of a computer, can open new possibilities and bring joy and delight to the user[20].

6.2 Cultural Metaphors
The designer must give extra care when designing metaphors. In case they design for a wide range of audience, they must be careful of cultural differences. For example, in some countries sharing images of anthropomorphic images breaks a cultural taboo[13]. This can make many users feel displeased with the product, and eventually reject it.

6.3 Do not compromise system functionality
Metaphors should fit to the functionality and not vise-versa[19]. Designers should not strip any functionality of the system because the metaphor does not match the system functionality. In that case a different metaphor should be used.

6.4 Be careful for unmatching metaphors
Composite metaphors can be the solution to better understanding complex system functions to the user, but they can also be catastrophic if not used properly. In case multiple metaphors need to be used in the same context, it might be wiser to be of the same domain.

6.5 Use only necessity
Sometimes metaphors can be used too much. As Barr suggests, do not use for the sake of using[2]. Too many metaphors can act negative to the user’s understanding of the interface as well.

6.6 Beware of Structural Metaphors
Structural Metaphors are focused on the user’s real world experiences and familiarity with existing objects. Because of this the expectations of the users about the capabilities of the object are high. If the signified object does not behave the same way the signer does, it can cause confusion and frustration to the user.

7. DISCUSSION
User-interface metaphors are a great tool to introduce new concepts and technologies to users. For new and novice users, a metaphor can be the guiding element to the user’s first step of technology. To more advances users, a metaphor can be a great boost into getting to know the functionalities of a program better. Once the user, regardless of skill, understands the how to use the interface, the metaphor serves no purpose. Also, the fact that metaphors do not scale or age well, along with the fact that metaphors can be culture specific, make user-interface metaphors hard to use. The dramatic drop in learning time can be too good to be easily ignore, but is it worth it?

What there be some sort of balance between the literalism of a metaphor and the abstraction of a computer system? Can a system be designed in a way that, it is as abstract as needed so the power of the computer is not compromised, but also the user learns through metaphors? Can a design exploit the user’s prior knowledge and be free of any metaphorical entailments? How liberate can a design be? What are the characteristics of the signifier object needs to be transferred to the simplified object in order not to confuse the user? What if we replace the metaphor with an abstract concept once the user is experienced enough? Guidelines or patterns for designing with metaphors should be created. Who knows what the future holds for the user-interface? Maybe in some years from now the desktop metaphor will be long gone and some new metaphor would take its place. Can metaphors co-exist in a abstract environment that looks nothing like anything from the real world? Would that confuse the users? If we can find a balance between using metaphors and using abstract concepts to visualise data and system functions, could we have the user friendliness of the metaphors without compromising any of the system’s capabilities?

8. CONCLUSION
Ever since the release of the first graphical user-interface, metaphors have been a great component of designing interfaces. They help us design interfaces that are instantly available to the users, by using their prior knowledge and experience about the world and different interfaces. Metaphors in interfaces can be a great way of introducing users to new technologies and system functionalities. They create a feeling of familiarity to the user and make the interface delightful and enjoyable. Interface metaphors can be portrayed like metaphors of different fields. They tend to have the same characteristics and be placed in the same categories and be used likewise. Interface metaphors can have negative effects if not used right. Like every physical object, digital object have affordances. In case some affordance of the simplifier object is missing, it can lead to confusion and anger. When designing with metaphors, the designer needs to be careful of the metaphoric entailment of the design. Metaphors need to be used careful in order to improve the experience of the users and not ruin it. Guidelines in designing metaphors should be created, for the consistent use of metaphors in interfaces. Some suggestions from various papers were collected as the beginning of guidelines on using metaphors.
9. REFERENCES


