List of Design Interventions

Deliverable 4.4a

Deliverable Title	List of Design Interventions
Deliverable Number	D4.4a
Authors	Casper Van Gheluwe (imec), Thomas De Meester (imec), Brett Binst (VUB-SMIT), Annelien Smets (VUB-SMIT), Evelien Marlier (imec)
Date	2023-05-31 (v1)

This document is part of the Serendipity Engine project, an interdisciplinary research project funded by FWO (Research Foundation Flanders) under grant number S006323N. The project duration is four years (2022-2026).

The Serendipity Engine project focuses on serendipity and recommender systems. While many advocate for designs for serendipity in recommenders, one could ask the question what does this mean in practice? Serendipity is generally understood as a beneficial design principle ought to deliver societal value, however, putting it into practice still presents major challenges. The Serendipity Engine project sets out to address these challenges and support societal stakeholders in designing recommender systems to foster serendipity in public contexts.

The Serendipity Engine research consortium consists of the following partners:

- Vrije Universiteit Brussel, **SMIT** (Studies in Media, Innovation and Technology)
- Universiteit Gent, **IDLab**
- Universiteit Antwerpen, **Adrem Data Lab**
- imec, **EDiT** (Enabling Digital Transformations)

For more information about the project, visit our website www.serendipityengine.be.

Contact details project coordinator:
dr. Annelien Smets
Vrije Universiteit Brussel
annelien.smets[at]vub.be
serendipityengine[at]vub.be

Table of Contents

Introduction

In today's rapidly evolving technological landscape, it is essential to explore innovative ways to utilize technology for the betterment of society. To achieve this goal, it is necessary to design and create prototypes of new technologies that can make a real difference in people's lives. The process of building these prototypes involves several stages, including design, creation, and evaluation, which require a deep understanding of the underlying technology.

To extract valuable insights from these processes, design science research is crucial. Design science research aims to develop theories and frameworks that enable researchers to systematically explore and evaluate the design and creation of prototypes. One such framework is the Information System Design Theory (ISDT), which provides a comprehensive approach to prototyping.

The purpose of this deliverable within the Serendipity Engine project is therefore to build knowledge on novel component designs that aim to engender serendipity in (web) applications through such ISDTs. We will design, create, and evaluate several prototypes of these app components within three pilot projects. By doing so, we aim to extract valuable insights that can be used to develop and improve similar app components in the future with serendipity in mind. This way, we ensure that the design and implementation work within the pilots is also applicable elsewhere, making it more usable and impactful.

Additionally, we will be leveraging the concept of affordances and affordance feature repositories. Affordances are the potential uses or actions that an object or system offers to its users. Affordance feature repositories are collections of affordances that can be used to inform the design of new systems. By using affordances and affordance feature repositories, we aim to ensure that our prototype is designed with the needs of its users and stakeholders in mind.

This deliverable will provide an overview of our approach, including how ISDT and affordances will be incorporated into the project. We will also discuss the ISDT framework and detail how we will utilize it to achieve our research objectives. Furthermore, we will also discuss the implementation of ISDTs in our pilot studies and the overall lessons learned from working with ISDTs. Finally, we will present our findings and conclusions, along with recommendations for future research.

Information System Design Theories

Effective information systems

Information Systems (IS) are complex socio-technical artifacts that involve people, technology, organizations, and societies. They are designed to support various human activities, such as communication, decision making, learning and entertainment. However, designing effective information systems is not an easy task. It requires a deep understanding of the problem domain, the user needs and preferences, the available technologies and their capabilities and limitations, the organizational and social contexts, and their implications for IS use and evaluation.

One way to facilitate the design of effective information systems is to use Information System Design Theories (ISDTs). An ISDT is a prescriptive theory that offers theory-based principles that can guide practitioners and scholars in the design of effective information systems and set an agenda for on-going research. They integrate normative and descriptive theories into design paths that specify how to achieve certain goals or outcomes with certain means or methods.

An ISDT is in essence a structured approach to information system design that involves a set of components, including a *kernel theory*, *meta-requirements*, *meta-design*, and one or more *testable design product hypotheses*. This theory is generally extracted by building a set of prototypes and evaluating them. Afterwards, the theory can be used to guide the design and implementation of a final information system.

Some of the fundamental elements that form ISDTs are loaned from existing concepts that are used in building architecture, such as design principles and design patterns. These concepts are used to create a set of high-level requirements, called meta-requirements, that define what the information system should do or provide for its users or stakeholders. These requirements are made by combining the needs of the users and the kernel theory. This kernel theory is a set of propositions derived from existing knowledge or theories that explain why a certain design will produce certain effects under certain conditions.

In the context of a research project, the design requirements should be innovative. Otherwise, there is no point in using ISDT. From the requirements, a backlog of user stories is formed, which is then used to design the features of the information system. These features are then collected and evaluated to form the design principles. These principles operate more on the conceptual level, describing how the information system should be constructed or configured to meet the meta-requirements.

Examples

Several authors have developed information systems by following the principles of an ISDT.

Jones and Gregor, for example, argue that the relational database design model that was put forward by Edgar F. Codd back in 1970 [1], can already be described as an information system design theory [2]. Codd's database model is based on a kernel theory that states that a well-designed database should be organized into tables with well-defined relationships between them. The meta-requirements for this model include the ability to store and retrieve data efficiently, maintain data integrity, and support concurrent access by multiple users. The meta-design principles for the model include the use of a relational schema, normalization of data to minimize redundancy, and the use of a query language to access and manipulate data. To test the design, the model includes a set of testable hypotheses that predict how well a database designed using the model will perform in terms of efficiency, scalability, and data integrity.

Another example of an ISDT is the vigilant executive information system (EIS), which is designed to provide real-time access to critical business information for executives [3] [4]. The kernel theory behind this model is that executives need access to timely and relevant information to make informed decisions. The meta-requirements for the system include the ability to integrate data from multiple sources, provide timely and relevant information, and support decision-making at the executive level. The meta-design principles for the system include the use of a dashboard interface that displays key performance indicators, the ability to drill down into data to get more detailed information, and the use of alerts and notifications to keep executives informed of critical events. To test the design, the model includes a set of testable hypotheses that predict how well the system will perform in terms of providing timely and relevant information, supporting decision-making, and improving organizational performance.

A third example of an ISDT is the learning-oriented knowledge management system (LOKMS), which is designed to help organizations capture, share, and apply knowledge to improve performance [5] [6]. The kernel theory behind this model is that organizations need to create a culture of learning in order to stay competitive in a rapidly changing environment. The meta-requirements for the system include the ability to capture and organize knowledge, facilitate sharing and collaboration among users, and support learning and continuous improvement. The meta-design principles for the system include the use of a knowledge repository that is accessible to all users, the use of social networking tools to facilitate collaboration and knowledge sharing, and the use of analytics to track performance and identify areas for improvement. To test the design, the model includes a set of testable hypotheses that predict how well the system will perform in terms of promoting a culture of learning, improving knowledge sharing and collaboration, and driving organizational performance.

Lastly, in 2015 Coenen *et al.* have presented an ISDT for information systems that are used for the assessment of human competences while supporting learning [7]. They first introduce the requirements that originate from the current state of practice in human evaluation. Afterwards they present several design principles that can address these requirements. These principles are then concretely instantiated by a number of design features, that can be put into a working prototype. These outputs together form an ISDT.

(Dis)advantages

Using ISDT provides several advantages for information system design. One the one hand, it provides a structured and systematic approach that helps designers to plan, analyse, design, implement, and maintain the system in an efficient and well-organized manner. Secondly, it enables designers to create more efficient and effective systems by identifying and eliminating redundancies and inefficiencies. Using an ISDT also refocuses the developers on designing systems that are user-friendly and easy to use, resulting in enhanced usability and user satisfaction. Fourthly, by following a structured approach to design, ISDT can help reduce the development costs of information systems as fewer re-developments will be required. Finally, thanks to its focus on structured documentation and hypothesis testing, systems (or components thereof) designed via an ISDT can be easier to reuse when new use cases are developed, even if the scope is widely different.

However, there are also some disadvantages to using ISDT. Firstly, some design theories can be overly rigid and inflexible, which limits the ability of designers to create unique or innovative systems. Secondly, some ISDTs, and even the concept of an ISDT itself, can be complex and difficult to understand, particularly for non-technical stakeholders, resulting in confusion or misunderstandings. Thirdly, implementing it can require a significant investment of time, resources, and expertise, which may not be feasible for small organizations or projects with limited budgets. Finally, the structured approach to design required by this approach can be time-consuming, particularly in the planning and analysis phases, which may not be acceptable for projects with tight deadlines.

An ISDT for the Serendipity Engine

One of the challenges of designing applications and recommender systems to foster serendipity in public contexts, such as for the tourism and entertainment industry or for sustainable mobility, is to provide a sound theoretical framework that can guide and justify the design decisions made for different scenarios and stakeholders. As we argued in the previous section, an ISDT is in fact such a prescriptive theory that offers theory-based principles to can guide practitioners in the design of effective information systems. As mentioned, it consists of a set of constructs, models, methods, and instantiations that describe the problem domain, the design process, and the design product of an information system.

Within the Serendipity Engine project, we can benefit from the use of an ISDT to thoroughly describe the design decisions that were made over the course of the three pilot projects. An overarching ISDT structure can help to articulate the goals, requirements, and assumptions of each pilot, as well as the design principles, methods, and artifacts that are used to achieve them along the way. Each pilot project will start from its own hypotheses and its own goals, but the ISDT, and therefore this document, will enable the researchers to record the design approach and the lessons learned from using these approaches in a structured way. Thus, the developed design theory can facilitate the communication and dissemination of the design knowledge generated by the project to other researchers and practitioners who are interested in designing information systems for serendipity.

Moreover, an ISDT could also be used to map how several contextual factors affect the experienced serendipity when using the prototypes that will be developed over the course of the project in collaboration with the pilot partners. Examples of such factors could include the weather (i.e. cyclists may be less willing/likely to take a detour to experience a new area during a rain storm), the COVID-19 crisis (i.e. visitors may not be able to visit all the historic landmarks in a city when they are closed because of quarantine measures), or the level of inflation (i.e. higher ticket prices may deter potential visitors from attending a concert of an artist they do not yet know).

As the project runs, and the three pilots are developed and activated, the Serendipity Engine ISDT will be co-developed by the project partners, together with the pilot partners and other stakeholders, and the document will be updated as necessary.

User experience design

Designing for Serendipity

Designing for serendipity is an intentional design approach that aims to create opportunities for unexpected and valuable discoveries by creating experiences and products that encourage exploration, curiosity, and learning.

While design patterns establish solutions to common design challenges, serendipity allows us to focus on exploring new and unexpected possibilities. The design patterns provide the structure (familiarity) in which creativity, chance, spontaneity, and experimentation can be encouraged. The Greek philosopher Heaclitus (544 – 484 B.C.) argued "*If you do not expect it, you will not find the unexpected, for it's hard to find and difficult*". This implies that we first must meet the user's expectations and organise the information accordingly, before we can evolve to inducing serendipitous experiences.

McCay-Peet and Toms categorize serendipitous encounters in 3 types [8]:

1. **Type A:** from observation to a solution. This happens when people are not looking for particular information. A great example of this is scrolling a social media's feed.
2. **Type B:** from problem 1 to a solution to problem 2. This happens when people are looking for specific information, but instead encounter information for another (unsolved) problem.
3. **Type C:** unexpected solutions. This happens when an individual is looking for information, but accidentally find an unrelated, and thus unexpected, information to a solution.

Designing for serendipity might sound as an oxymoron, as it is a subjective experience in which the unexpected and unpredictable plays a crucial role but could in fact be artificially cultivated by creating opportunities for it through the purposeful design of the physical or digital environment [9].

The following four guidelines can serve as inspiration when designing a product that engenders serendipity:

- Incorporate elements of randomness, which makes room for unexpected outcomes and presents users to new content they would otherwise not have discovered [8];
- Encourage exploration so that the user interacts with the product in unexpected ways and discover new features or content, curiosity plays a key role in increasing the chance of discovery [10];
- Allow for personalization to guide the discovery of new elements based on their personal preferences, which in turn leads to a higher chance of a serendipitous moment [8];

- Embrace ambiguity and leave room for interpretation so that the user is encouraged to find out more, explore and experiment [11] [12];

Design principles

Design principles are a set of fundamental guidelines that designers use to create usable, and aesthetically pleasing designs. These principles help designers make decisions about various design elements, such as layout, form, colour, component and more.

The design principles are based on the *Gestalt theory* of visual perception, a psychological theory that describes how people perceive and organize visual information. The Gestalt theory originates from the field of psychology in the early 1900's, where German psychologists Max Wertheimer [13], Kurt Koffka [14] and Wolfgang Kohler [15] looked for ways to understand how humans (and infants) perceive and give meaning to visual stimuli around them. The word "gestalt" refers to the German word for shape or form (in Dutch "*gestalte*"). The theory suggests that people (the viewer) seek a "gestalt" or unified whole [16]. The viewer is looking for a connection between the elements, for some form of organization and by doing so, tends to perceive visual elements rather than looking at the individual components. The Gestalt theory provides a framework for understanding how people perceive and make sense of visual information.

Designers use these laws of visual perception, as design principles to organize and bring structure to their work and use these principles to prevent the user from being overwhelmed with information.

These principles help a designer:

- Determine which design elements are effective.
- Direct a viewer's attention to act or steer their behaviour.
- Design products or services that meet the user's need and solve their problem.

By applying and combining the design principles listed below, it is possible to create and manage these design patterns and further analyse, optimize, and maintain them. A good implementation creates information systems that support the business goals, improve the quality of the product or service, enhance the user experience, and foster an environment of creativity and exploration which might lead to serendipitous discoveries.

List of design principles

Balance

Balance is the psychological sense of equilibrium. As design principles, balance places the parts (elements of the visual design) in an aesthetically pleasing arrangement.

- **Symmetry:** the parts are placed identically on both sides of the design and have equal weight on both sides.
- **Asymmetry:** the parts are placed unequally and/or with different weights, but they still appear balanced.
- **Radial:** the parts are arranged around a central point.

Asymmetric designs are more dynamic and informal and normally keep the viewer's attention. As such, asymmetric designs (or part of a design) could be a way to catch the viewer's attention and induce serendipity.

Unity

The principle of unity (in the Gestalt theory this is referred to as the law of Prägnanz) is the fundamental principle of design and supported by all other principles. If a design is not unified, it cannot be considered successful or effective. Unity allows each individual element in a design to coexist with another element and form a harmonious whole, both visually and conceptually.

Proximity

Related items are grouped together. A viewer will perceive elements that are close together as a group, and of similar meaning. Even when the shape, size and colour of the elements are radically different, they will appear as a group if they are close together.

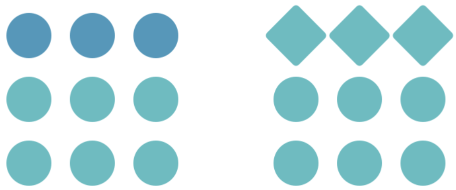
Proximity, the space between elements, allows the designer to organize elements (group or separate) that are visually related.

Take for example a toolbar, it consists of a set of buttons that share characteristics. By grouping and separating the buttons, we can organize the toolbar by function.

Similarity

Elements that share visual characteristics such as shape, size, colour, texture, value, or orientation will be perceived as belonging together.

A viewer tends to perceive similar elements as a pattern: they serve the same purpose. Similarity helps the designer organize and classify visual elements. Visual weight or contrast allows us to differentiate between similar elements and draw the user's attention.



Continuation

Continuation means that something (a line, an edge, a curve, a direction) continues from one element to another. The viewer's eye will follow the continuation from one element to the other and group the elements because of that. It unifies different groups or elements into a larger whole and helps to improve the page's scan-ability and legibility.

Disruption of this line continuation can signal the end of a section or draw the user's attention to a new piece of content. Continuation can also be strategically used to communicate affordances and signifiers, for example in carousels where only a part of the image is visible, giving the viewer the impression that they need to swipe.

Alignment

Alignment is an organizing principle that arranges the elements to indicate a direct relationship. So, while proximity can push items apart, alignment can tell us they still belong to the same piece, creating structure of related elements with different meaning.

Repetition

Repetition is based on grouping by similarity; elements that are similar visually are perceived to be related. Any element can be repeated - line, font type, shape, colour, value, or texture -as well as other things such as direction, angle, or size. Repetition helps unify a design by introducing patterns.

The repetition of some aspects of the design throughout the entire piece helps create a unique and consistent design, with recognizable elements (for example certain components, brand identity, etc..).

Figure-ground

Figure and ground refer to the perception of objects either being in the foreground (figure) or background (ground). This determination happens subconsciously.

This principle is used all over a design, but for example is put to practice in dialogue boxes, where the content becomes secondary when the modal appears.

Emphasis and dominance

Emphasis is the part of the design that catches the viewer's attention. Elements that break visual hierarchy are considered dominant.

Contrast in design is an *accentuation* of the differences between elements in a design. Contrast occurs when two similar elements on a page are different.

- Contrast is attractive to the eye and grabs attention. This improves scan-ability of the page.
- Contrast aids the organization of information and creates visual hierarchy.
- Contrast creates focus and emphasis.

This principle allows us to create focal points that capture and hold the viewer's attention.

Closure

Closure is the principle that viewers perceive incomplete objects as a whole. This tendency of grouping visual elements that (partially) match as one object, helps the viewer convert complex objects into simpler or known shapes.

This principle is often used in logo and icon design, where white space is used to make a few simple shapes appear as a complete shape. Closure can also be used to signal additional content, when only parts of the item are shown. This incompleteness indicates they should scroll or swipe.

Design patterns

When using a product or service for the first time there is likely to be an element of learning. This learning curve can be an uncomfortable experience, especially when the design feels unfamiliar. By matching the steps, layout and information with the expectations and prior experiences of the user, this friction is reduced.

Matching the viewer's expectations is achieved by using common conventions, patterns, applying the design principles and by maintaining a consistent design throughout the application. When we apply and design for a consistent and predictable experience, we are using design patterns. Design patterns create a consistent and repeating visual language within a design.

By using the same design elements, such as colour, shape, weight, or font, in a deliberate and repetitive way, a sense of unity and cohesiveness is created that helps to recognize the organization, navigate the information, and reinforce the message or purpose of the design.

Patterns are a general repeatable solution to a commonly occurring problem. It is a visual description or template for how to solve a problem that can be used in many different situations. Patterns allow user to recognize what certain design elements are meant to do and what to expect, based on prior experiences.

Jakob Nielsen describes this tendency for users to expect and behave based on their cumulative experience from other websites, in a law, "Jakob's law", of internet user experience [17]. The law states that users prefer to use a website or application in a similar way that they'" accustomed to use other websites that they already are familiar with. We leverage this prior knowledge (mental model) by applying existing patterns, allowing the user to focus on their tasks rather than interpreting the interface.

Patterns make it intuitive and easy for the user to engage with the design, since no learning curve is required, which in turn guides the user's behaviour and improves overall experience. A bonus of keeping a consistent visual language is that by breaking the consistency, we indicate to users where things might work differently than expected.

However, not every design challenge has a readily available design pattern to solve it. In these cases, we look to affordances and signifiers to help guide the user in their interaction with the (unfamiliar) design.

List of design patterns

The following guidelines are important to create easy-to-use and intuitive designs:

Discoverability

Discoverability refers to how easy content is to find, but as well whether it's possible to figure out how to use an object by interacting with it. For Don Norman, discoverability embodies all crucial concepts of design [18]:

- Affordances: the perceived properties of the object
- Signifiers: the cues that communicate the affordance of the object
- Constraints: the imposed or perceived limitations of the object
- Mappings: the organization of the design and the relationship between objects
- Feedback: the communication of an action by the system

Visibility

Information and functions should be easy to discover. By using clear and consistent affordances and signifiers, it's easier for users to find and use the important features of a product or service.

Feedback

In his book The Design of Everyday Things, Don Norman states that "*a lack of feedback creates a feeling of lack of control, which can be unsettling*" [18], implying that feedback is a crucial element in the communication between user and system. Providing feedback is important to help a user understand how to interact with objects and what effect their action has on the current state of the object or system. Feedback should be used to prevent errors, and when they occur, detect, and help the user recover from them quickly.

Consistency

Consistency helps us create predictable, cohesive, and intuitive designs and introduce new, easy to learn, patterns. In his book Usability Engineering, Jakob Nielsen states that websites should maintain internal consistency by using a consistent user interface for their product or product family and follow established industrial conventions (external consistency) [19].

Simplicity

The range of possibilities should be no more than necessary.

The pareto principle (also known as 80-20 rule) in software design and development is the rule that 20% of the functionality is used 80% of the time [20]. Therefore, we should make the most common and important functions easiest to find. We can do this by hiding or reducing the prominence of infrequently used or advanced functions. Simplicity helps to reduce the cognitive load and streamline the user experience, while at the same time allowing for sufficient ambiguity for the user to explore.

Affordances

Affordances are the perceived properties of an object or environment that suggest or determine how they can be used [18]. Signifiers are the sensory (visual, auditory, tactile) cues that communicate the presence of an affordance. For example, a button affords pressing, the icon inside might signify to the user that it will start the video (a play icon).

By making use of recognizable affordances, we try to create intuitive and easy-to-use designs that make it clear how to interact with them, even when the interaction itself is new to the user. When no affordance is readily available, we experiment with different cues to create new affordances. For example, if a new type of interface control is being used, such as a gesture-based control, designers may need to experiment with different visual or auditory cues to help users understand how to use it.

Well-known examples in human-computer interaction are 'click*able*' buttons or 'dragg*able*' sliders. In that sense, Dourish' definition of an affordance "as a three-way relationship between an environment, a human being, and a potential activity" [21, p. 118] is particularly comprehensible and depicted in Figure 1.

Figure 1 - Three-way relationship between the environment, a human being and an activity [22].

An “affordance approach to serendipity” has been proposed [23] meaning that serendipity can be considered an outcome of an environment-actor correspondence. This aligns with Foster and Ellis concluding that serendipity “does not exist within a vacuum, it is the product of context”. Following Björneborn’s usage of the terminology [23], we refer to the ‘capacities’ of environments, ‘capabilities’ of actors, and in this case ‘experiencing serendipity’ as the possible afforded activity through their interplay (see Figure 2). Serendipity then occurs when there is an “actualization” of the affordance [24]. In other words, if – not ‘when’ because it is a “potential” [23, p. 1066] – there is a “complementarity of the acting organism and the acted-upon environment” [25, p. 2]. Indeed, the interaction between actor-environment is possible, but not necessary. It is not because a button is clickable, that there is an actor clicking on it.

Affordance features

In terms of designing affordances, the notion of “affordance features” becomes relevant. These are “the structural elements of artifacts that provide affordances” [26, p. 2]. Such features can be selected from a broader range of possibilities. This is what Kim et al. refer to as “affordance feature repositories” that consist of “multiple affordance features for a given affordance” [26, p. 2]. In other words, an affordance feature repository consists of affordance features that can provide clues to design new affordance features through analogical reasoning [26].

The relationship between these different concepts is illustrated in Figure 2. Here, the key message is that there exist multiple ways to design for affordances, which are captured in an affordance feature repository. The selection of particular affordance features (or short: features) is then guided by design intent, which also incorporates knowledge about the specific domain and context. For example, when the door is supposed to be placed in a children’s toy house, the door handle will have to be one that is manageable by children and thus rather one that can be pushed instead of a latch that needs to be opened.

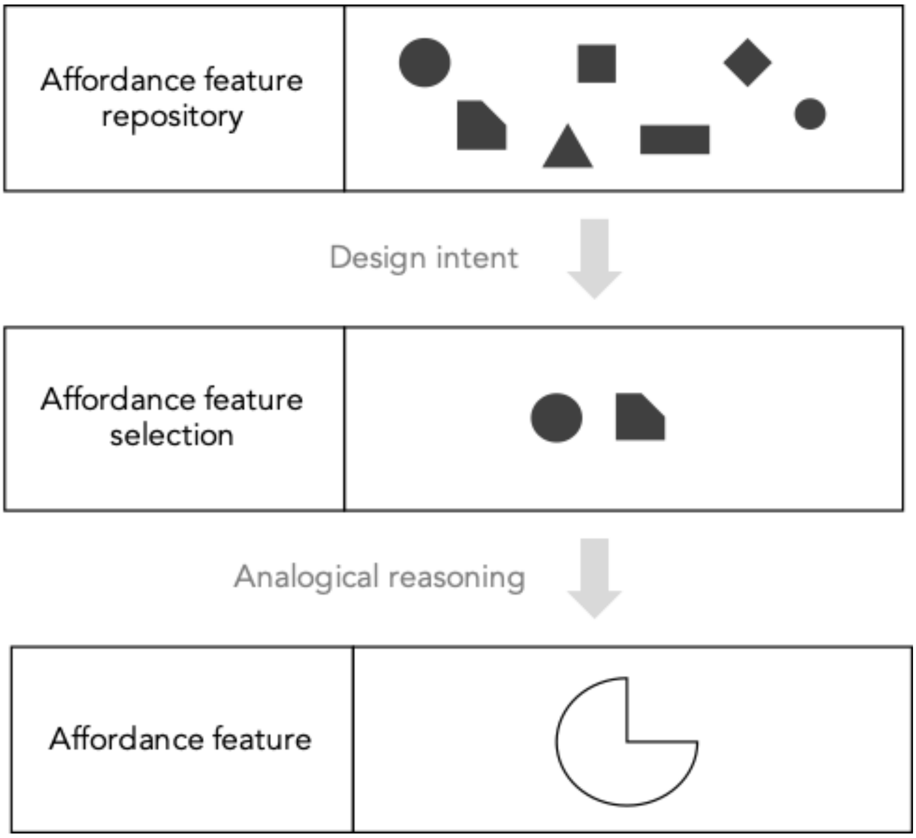


Figure 2 - Affordance feature repositories consist of a collection of multiple features that contribute to an affordance [22].

Affordance feature repository

The affordance feature repository contains a set of features that might afford serendipity. The goal of the repository is to:

(1) Provide a means to systematically gather research insights about the effect of these features. Therefore, it is important to store information about the domain in which it has been studied (e.g., mobility, book recommender, movies, etc.).
(2) Serve as a *'brainstorm tool'*: by providing a lot of examples (even from other domains) it might inspire developers to design a novel feature in their product. For example, a well-known means to achieve 'slowability' in physical environments is putting street benches [23]. Perhaps this might also be inspiring for online environments ("what would a 'street bench' look like on a website?").

In earlier work and building upon the affordance approach to serendipity [23] an affordance feature repository [27] has been proposed which will be further developed in the Serendipity Engine project.

Pilots

This document will be updated about the pilots being conducted in the Serendipity Engine project. We will provide information on how the IDST has been applied to the pilots.

Lessons learned

Based on the pilots this document will list lessons learned regarding designing for serendipity.

Conclusions

Conclusions and recommendations will be added here.

References

[1] E. F. Codd, "A relational model of data for large shared data banks," *Communications of the ACM,* vol. 13, no. 6, pp. 377-387, 1970.

[2] D. Jones and G. Shirley, "The anatomy of a design theory," *Journal of the Association for Information Systems,* vol. 8, no. 5, 2007.

[3] J. G. Walls, G. R. Widmeyer and O. A. El Sawy, "Building an information system design theory for vigilant EIS," *Information systems research,* vol. 3, no. 1, pp. 36-59, 1992.

[4] J. G. Walls, G. R. Widermeyer and O. A. El Sawy, "Assessing information system design theory in perspective: how useful was our 1992 initial rendition?," *Journal of Information Technology Theory and Application (JITTA),* vol. 6, no. 2, p. 6, 2004.

[5] D. B. Paradice and J. F. Courtney, "Learning-oriented knowledge management system," 2003.

[6] D. Hall, D. Paradice and J. F. Courtney, "Building a Theoretical Foundation for a Learning-oriented Knowledge Management System," *Journal of Information Technology Theory and Application,* pp. 63 - 84, 2003.

[7] T. Coenen, L. Coertjens, P. Vlerick, M. Lesterhuis, A. V. Mortier, V. Donche, P. Ballon and S. De Maeyer, "An information system design theory for the comparative judgement of competences," vol. 27, no. 2, pp. 248-261, 2018.

[8] L. McCay-Peet and E. Toms, "Researching serendipity in digital information environments," *Synthesis Lectures on Information Concepts, Retrieval, and Services,* vol. 9, no. 6, 2017.

[9] T. Race and S. Makri, Accidental information discovery: Cultivating serendipity in the digital age, Cambridge: Chandos Publishing, 2016.

[10] M.-H. Lee, D.-H. Kim and T.-J. Nam, "Understanding impacts of hidden interfaces on mobile phone user experience," in *Proceedings of the 7th Nordic Conference on Human-Computer Interaction*, 2012.

[11] W. Gaver, J. Beaver and S. Benford, "Ambiguity as a resource for design," in *International Conference on Human Factors in Computing Systems*, Ft. Lauderdale, Florida, USA, 2003.

[12] Holmboe and P., Serendipity as an Experiental Quality: How Serendipity can be Used as a Resource to Create Meaningful Experiences, 2017.

[13] M. Wertheimer, "Laws of organization in perceptual forms," in *A source book of Gestalt psychology*, Kegan Paul, Trench, Trubner & Company, 1938, pp. 71-88.

[14] K. Koffka, Principles of Gestalt Psychology, New York: Harcourt, Brace, 1935.

[15] W. Köhler, Gestalt Psychology, New York: Liveright, 1929.

[16] D. Todorovic, "Gestalt principles," *Scholarpedia,* vol. 3(12), no. 5345, 2008.

[17] J. Nielsen, "End of Web Design," Nielsen Normal Group, 2000. [Online]. Available: https://www.nngroup.com/articles/end-of-web-design/.

[18] D. Norman, The Design of Everyday Things, Boston: MIT Press, 2013.

[19] J. Nielsen, Usability Engineering, London: Morgan Kaufmann, 1993.

[20] J. Persson and E. Nicklasson, Pareto principle in software: Feature usage and software development in relation to the Pareto principle (Dissertation), Malmö: Malmö University, 2022.

[21] P. Dourish, Where the action is: The foundations of embodied interaction, Boston: MIT Press, 2004.

[22] A. Smets, Serendipity as a Shared Value in Urban Recommender Systems (PhD Thesis), Brussel: Vrije Universiteit Brussel, 2022.

[23] L. Björneborn, "Three key affordances for serendipity: Toward a framework connecting environmental and personal factors in serendipitous encounters," *Journal of Documentation,* vol. 73, no. 5, pp. 1053-1081, 2017.

[24] L. Björneborn, "Adjacent Possible.," *The Palgrave Encyclopedia of the Possible,* pp. 1-12, 2020.

[25] W. Gaver, "Technology affordances," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Reaching through Technology*, Chicago, 1991.

[26] Y. Kim, J. Noh, S. Kim and others, "A case study for application of design for affordance methodology using affordance feature repositories," in *Proceedings of the 19th International Conference on Engineering Design (ICED13) Design for Harmonies*, Korea, 2013.

[27] A. Smets, L. Michiels, T. Bogers and L. Björneborn, "Joint Workshop on Interfaces and Human Decision Making at RecSys 2022," in *CEUR Workshop Proceedings*, 2022.