Towards a Pattern Language for Describing Distributed Interactions

Hajutatud interaktsioone kirjeldav mustrikeel

Master Thesis

Eduardo Mercer
emercer@tlu.ee
Author

Ilya Shmorgun, David Lamas
{ilja.smorgun,david.lamas}@tlu.ee
Supervisors

Institute of Informatics
Tallinn University
Narva Rd. 29, 10120 Tallinn
Estonia

Author: ....

Supervisor: ....

Supervisor: ....

Head of the Institute: ....

TALLINNA ÜLIKOOL

Non-exclusive licence to reproduce a thesis and make thesis available to public

I, *Eduardo Ferreira Fontes Mercer* (date of birth: *17/03/1979*)
(*name of author*)

1. grant Tallinn University a permit (a non-exclusive licence) to reproduce for free and make public in the repository of Tallinn University Academic Library a piece of work created by me
   Towards a pattern language for describing ubiquitous interactions
   (Kõikjaleulatuvaid interaktsioone kirjeldav mustrikeel),
   (*title of thesis*)

   supervised by *Ilya Shmorgun and David Lamas*.
   (*names of supervisors*)

2. I am aware of the fact that the author also retains the rights mentioned in Clause 1.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights of other persons or the rights arising from the Personal Data Protection Act.

Tallinn, 4/05/2015

(digital) signature and date

Summary

Building upon previous literature review on the design of distributed user interfaces, this thesis traces an attempt to collect all instances of interaction techniques targeted towards Multi-Display Environments into design patterns, which are general reusable solutions to problems unique to the area of MDEs. Once these problems are properly identified and their solutions collected into a pattern catalogue, the study expands into applying taxonomic and social network analysis to identify how these solutions relate between themselves and build the scaffolding for a pattern language for distributed user interfaces.

Eestikeelne kokkuvõte

Põhinedes eelnevale hajus-kasutajaliideste disaini uurivale ulatuslikule kirjandusele jälgib antud teadustöö katset koodadada kõiki juhtumeid, kus interaktsioonitehnikaid on suunatud multi-kuvari keskkondade disaini mustritesse - üldistesse taaskasutatavatesse lahendustesse probleemidele, mis on unikaalsed multi-kuvari keskkondadele. Pärast probleemide kindlaksmääramist ja lahenduste mustri-kataloogi kogumist laieneb uurimus rakendades taksonoomilist ja sotsiaalvõrgu analüüsi, et identifitseerida, kuidas need lahendused üksteisse suhtuvad ja ehitada alus hajus-kasutajaliideste mustrikeelele.

Table of Contents

1. Introduction

Melchior (2011) defines Distributed User Interfaces as an area of Human-Computer Interaction studying interfaces distributed across various displays, devices, and users engaged in co-located or remote collaboration.

This can be as simple an arrangement as one user and two devices (think of the growing user-base of smartphones putting computer users into a situation of possessing two computing platforms) or a complex working environment spanning public displays, personal computers, and private devices (tablets, smartphones, smartwatches) of several people working on concurrent networked tasks (think of a modern open-plan office with lax Bring Your Own Device policies).

These arrangements bring new challenges, not only of an organizational kind, but also those that interest the science of HCI, such as how to interact and collaborate on data living 'in the cloud' being handled through a multitude of devices with differing interface capabilities.

According to Elmqvist (2011), one of the main challenges of DUI is a lack of generalizable models, frameworks, and toolkits that support DUI development. Shmorgun & Lamas (2014) surveyed the existing literature for DUIs in 2014 in search of existing methods, approaches, and challenges in the design of DUIs.

It agrees with Elmqvist that most the work on DUI design is focused on either explaining existing interactions or proposing new ones, but few are generalizable outcomes enabling designers to understand the possible options for going from a design concept to tangible artifact while matching the assumptions of the users. It also identifies a need for concrete examples of how such conceptual outcomes can be applied in a specific design case and what that could result in.

Our research problem, thus, was figuring out how to provide interaction designers with a means of bridging the gap between design concept and tangible artifact.

Of a corpus of 105 articles surveyed by Shmorgun & Lamas (2014), 35 described one or more interaction techniques applied to multi-display environments. This outlines our primary research goal: taking these existing interactions techniques and building a catalogue of design patterns - reusable solutions to recurring design problems, not in the form of code snippets, but as templates or descriptions on how a problem has been solved in the past.

Design patterns have become a common practice in computer science since Gamma et al. - commonly referred to as the "Gang of Four" - published *"Design Patterns: Elements of Reusable Object-Oriented Software"* (1995), but they originated with a book by architect Christopher Alexander called *"A Pattern Language: Towns, Buildings, Constructions"* (1997).

According to Dearden & Finlay (2006) the main difference between Alexander's pattern language and the GoF's pattern catalogue is that the GoF's patterns show some interrelationship, but stop short of forming a language in Alexander's sense, where every pattern comes from other patterns and leads to yet another set of patterns in a hypertext where the relationship between patterns is as much part of the language as the patterns themselves.

A secondary goal of this research was to, after cataloguing all patterns, establish how they connect to each other in an attempt to build a pattern language out of this catalogue.

The research questions we tried to answer through this exploration were what are the recurring problems in distributed interaction design and which design patterns represent proven solutions to these identified design problems. We believed that throughout the collection and construction of a pattern language we would been able to answer these questions.

2. Procedure

2.1. Design Pattern Structure

A Semantic MediaWiki was used to collect all interaction techniques previously identified by Shmorgun & Lamas (2014). Semantic MediaWiki is an extension of MediaWiki, the same software that enables the existence of Wikipedia for example, but expands its content management capacity through *semantic annotations*, which allow machine processing of the data within. These annotations make it compatible with the Semantic Web, a W3C collaborative effort in making data in web pages shareable and reusable across different applications through a common framework. This allows any researcher to explore new relationships in the pattern catalogue using the Resource Description Framework (Krötzsch, 2006) instead of resorting to manual data re-entry.

The catalogue's wiki can be found at `http://idlab.tlu.ee/patterns/`

Using an extension called Semantic Forms, the following characteristics of each interaction technique were gathered into the wiki:

Summary

A simple paragraph describing how the interaction technique works. A quick reference for whoever is browsing the catalogue to see if this fits within their design problem without needing to read through the entire pattern.

Description

A more detailed explanation of the technique as presented in the original research, describing all the gestures involved and their results. Allows the reader to implement their own crude facsimile of the original technique without going into coding and implementation details (which can be found in the original research instead).

Reference

Points to the original article where the interaction technique was first described, allowing the reader to further explore its context and implementation.

Design Motivation

Collects the primary motivations guiding DUI design as identified in the literature review: Creating technological infrastructure, Augmenting existing practices, Designing new types of interactions, Creating engaging experiences, and Supporting the design process.

Design Goal

Collection of the main goals for designing DUIs, as identified in Shmorgun & Lamas (2014): Creating integrated workspaces, Improving information management across devices, Unifying the advantages of different devices, Fostering collaboration, Supporting joint interaction with information across devices, Supporting interaction in a free manner, and Supporting the design of interfaces for dynamic collections of devices.

Enabling technologies

List of DUI-related technologies that are applied in one or more of the interaction techniques surveyed: Displays, Multimedia, Alternative forms of input, Low-power high-performance processors, Networking technologies, Web technologies, Sensors, Physical object identification, Haptics, Machine-readable data formats and Databases.

Setting

This takes into consideration the context of use:

- *private*: single-person interaction with two or more devices;
- *semi-private*: interaction between devices in a small gathering of users;
- *public*: interactions in a meeting room, office environment or other public space where devices can be used to share information with a large group of people.

Example

A real-life implementation of the interaction technique, extracted from the original research or, in cases where broadly available, from other sources.

Diagram

A visual representation of the interaction technique.

2.2. Creating a Pattern Language

According to Fincher & Windsor (2000) there are four principles that should guide a pattern language:

- It should have a taxonomy so a reader can find patterns;
- It should allow readers to navigate to related patterns;
- It should allow for evaluation of problems from different standpoints;
- It should be generative, allowing users to develop new solutions.

The information collected in our MediaWiki was enough to describe individual design patterns, but fell short of a true pattern language. This was due to the relationships between patterns not being presented - it lacked, therefore, both a taxonomy and the means to navigate to proximal patterns. To enable the transition from catalogue to language, we needed to supply these tools.

In Fig. 1 it is possible to see a pattern as extracted from the wiki. The fields in bold - **Cites**, **Cited by** and **Related to** - required us to understand the taxonomy of out pattern catalogue; the path to which will be detailed in **sections 2.2** to **3.**

Bumping

Summary	Two tablets can be bumped together to trigger dynamic tiling of their displays. Picking up a tablet disconnects the displays. Holding the tablets at an angle during the bump triggers pouring data from one device to another.
Description	Hinckley (2003) proposes the use of a bumping gesture to dynamically connect together two tablet computers and form an extended screen area from the screens of individual devices when they are positioned next to each other. Removing one device from proximity reverts both screens to their previous individual state. The devices go into screen extension mode if both of them are resting on a desk. The same gesture triggers information transfer if both devices are instead being held. Bumping both devices together results in mutual sharing of information, however having one device slightly tilted during a bump results in one-way sharing of the titled device's clipboard to the receiving device.
Design motivation	Designing new types of interactions, Creating engaging experiences
Design goal	Creating integrated workspaces, Improving information management across devices, Supporting joint interaction with information across devices, Supporting interaction in a free manner, Supporting design of interfaces for dynamic collections of devices
Device type	Private, Semi-private
Enabling technology	Displays, Networking technologies, Sensors
Foreseen usage location	Private, Semi-private
Reference	Hinckley, K. (2003, November). Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (pp. 149-158). ACM. doi:10.1145/964696.964713 [7]
Cites	**Pick-and-Drop [1], Hyperdragging [4]**
Cited by	**Stitching [10], transSticks [14], Ubiquitous Graphics [15], F-Formations [23], DisplayStacks [25], EasyGroups [26], MobiComics [27], PaperVideo [29], Pinch [30], Conductor [35]**
Related to	**Hyperdragging [4], Stitching [10], transSticks [14], DisplayStacks [25], EasyGroups [26], MobiComics [27], PaperVideo [29], Pinch [30], Conductor [35]**
Diagram	

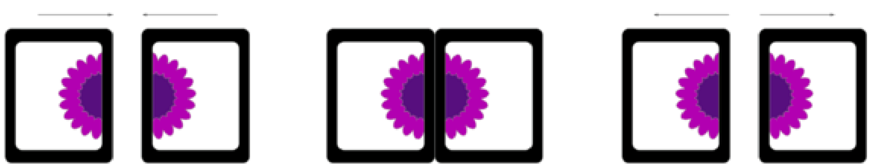


Fig. 1. Example of pattern from the catalogue

2.3. Taxonomic survey

Dearden & Finlay (2006), in their survey of HCI design languages, identify three possible relationships between patterns: derivation - where one pattern inherits elements from a higher level pattern, aggregation - where one pattern is contained within another pattern, and association - where one pattern uses another.

This leads to two types of relationship organization between patterns: either a pattern *enables* other patterns or it *completes* other patterns. If we were to analyze how one article describing an interaction technique cites and is cited by other articles in the same corpus, we would find the same type of relationships: earlier articles enable future articles citing it, further research by the same team completes past research and so forth. We decided, then, to base our taxonomy on bibliographic citations.

To do so, we trawled the ACM Digital Library search (available at `http://dl.acm.org`) for each of the articles singled out in Shmorgun & Lamas (2014). For each article, the ACM references list was then searched for the titles of all articles that had been published prior to it. Alternatively, Google Scholar (`https://scholar.google.com`) was used when an article could not be found in the ACM Digital Library catalogue.

The resulting survey of all 595* permutations can be found in Fig. 2. For clarity, the numbers used to label each article throughout this analysis match their citation numbers in this paper.

**the total number of cross-citations in a corpus of* n *articles is the triangular number* T_{n-1} *which is calculated through the formula* $\frac{(n-1) \times n}{2}$. *This is a case of the* handshake problem *and can also be calculated by the longer arithmetic divergent series 1+2+3+...+35 = 595.*

		Articles cited																		
		1	2	3	4	5	6	7	8	9	10	13	14	15	17	20	21	23	24	25
Shuffle, Throw or...	2	✓																		
Hyperdragging	3	✓	✓																	
ConnecTables	4	✓		✓																
That one there	5	✓																		
Drag-and-pop/-pick	6	✓	✓	✓																
Bumping	7	✓			✓															
SyncTap	8	✓		✓		✓														
Touch-and-Connect	9	✓																		
Stitching	10	✓		✓	✓	✓		✓	✓											
Interface Currents	12		✓				✓													
The Vacuum	13	✓	✓				✓				✓									
tranSticks	14	✓				✓		✓	✓	✓										
MuitiSpace	15			✓				✓	✓											
Ubiquitous Graphics	16			✓																
E-conic	17			✓							✓									
Select-and.Point	18	✓											✓	✓						
Slurp	19	✓																		
Chucking	20	✓	✓			✓	✓				✓	✓			✓					
Codex	21				✓						✓									
Lift-and-Drop	22	✓																		
F-Formations	23							✓						✓		✓	✓			
Conduit	24	✓		✓	✓				✓		✓						✓			
DisplayStacks	25				✓			✓			✓						✓			
EasyGroups	26							✓	✓		✓		✓							
MobiComics	27							✓	✓		✓									
MobiES	28			✓							✓									
PaperVideo	29	✓						✓												
Pinch	30				✓			✓			✓									
Seamless Interaction	32			✓											✓					
Cross-Device Drag...	33	✓		✓																
VisPorter	34	✓												✓				✓		
Conductor	35	✓		✓				✓			✓							✓		✓

Fig. 2. Initial Relationship Matrix. Rows represent citing papers, columns represent cited papers, citations are ticked. Papers with no citations skipped for brevity.

2.3.1. Social Network Analysis

The relationship matrix was then imported into Gephi, an open-source social network analysis tool. This allowed us to run several Force-directed graph drawing algorithms on the dataset to visualize the relationships between articles.

The purpose of these algorithms is to position the nodes of a graph (in our case, articles describing interaction techniques) in space so that all edges (citations between articles) are of more or less equal length and there are as few crossing edges as possible. This is done by assigning forces among the set of edges and the set of nodes, based on their relative positions, and then using these forces either to simulate the motion of the edges and nodes or to minimize their energy. Being physical simulations, these algorithms allow visualizing social network graphs without applying any complex graph theory transformations to the data.

Jacomy et al. (2014) compare three different algorithms; Fruchterman-Reingold, one of the pioneering algorithms in the area, described by TMJ Fruchterman and EM Reingold in 1991; Yifan Hu, created by Dr. Yifan Hu for Wolfram Research in 2005; and their own, ForceAtlas2, the default algorithm in Gephi.

The main strategic difference between the three is that Fruchterman-Reingold - being too resource-consuming - does not iterate, running once then stopping. Yifan Hu is able to run at variable speed thanks to its analysis of the variation of global energy in the graph; it therefore cools down after a while and settles to an arrangement. ForceAtlas2 runs continually and never reaches a cooldown, but can be interrupted by the user at any time.

We ran the same three algorithms on our dataset and the results can be seen in Figs. 3 to 5.

Fig. 3. Articles and citations as organized through Fruchterman-Reingold. Arrows represent citations between articles.

Colors were added later to represent our current understanding of how the articles are grouped. It's fairly evident that Fruchterman-Reingold was unable to group the articles in any fashion, instead just gravitating the largest nodes towards the center and the smaller ones to the periphery. Nodes are scaled according to their In-Degree, which is the amount of incoming edges connected to it or, in our dataset's case, the amount of citations an article receives from other articles in the corpus.

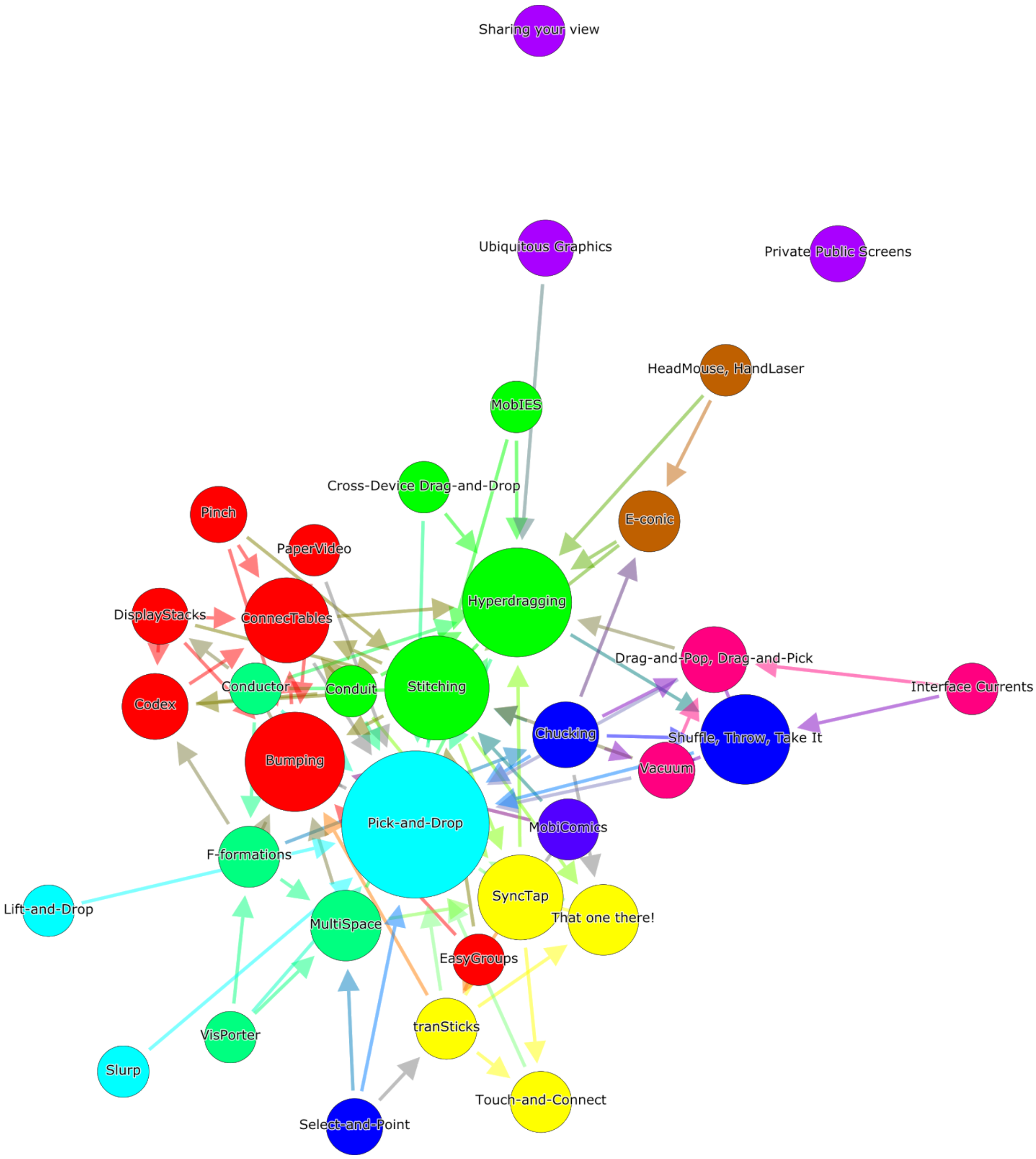


Fig. 4. Articles and citations as organized through Yifan Hu.

After cooldown, two unconnected nodes had drifted completely away from the cluster. Groups are a lot more evident than on Fruchterman-Reingold, but there are still too many articles belonging to unrelated groups in the same clusters while related articles are sometimes too far away. Also, the lack of repulsive forces clusters strongly-connected nodes too close together - this makes the edges too difficult to make out, making citation analysis cumbersome.

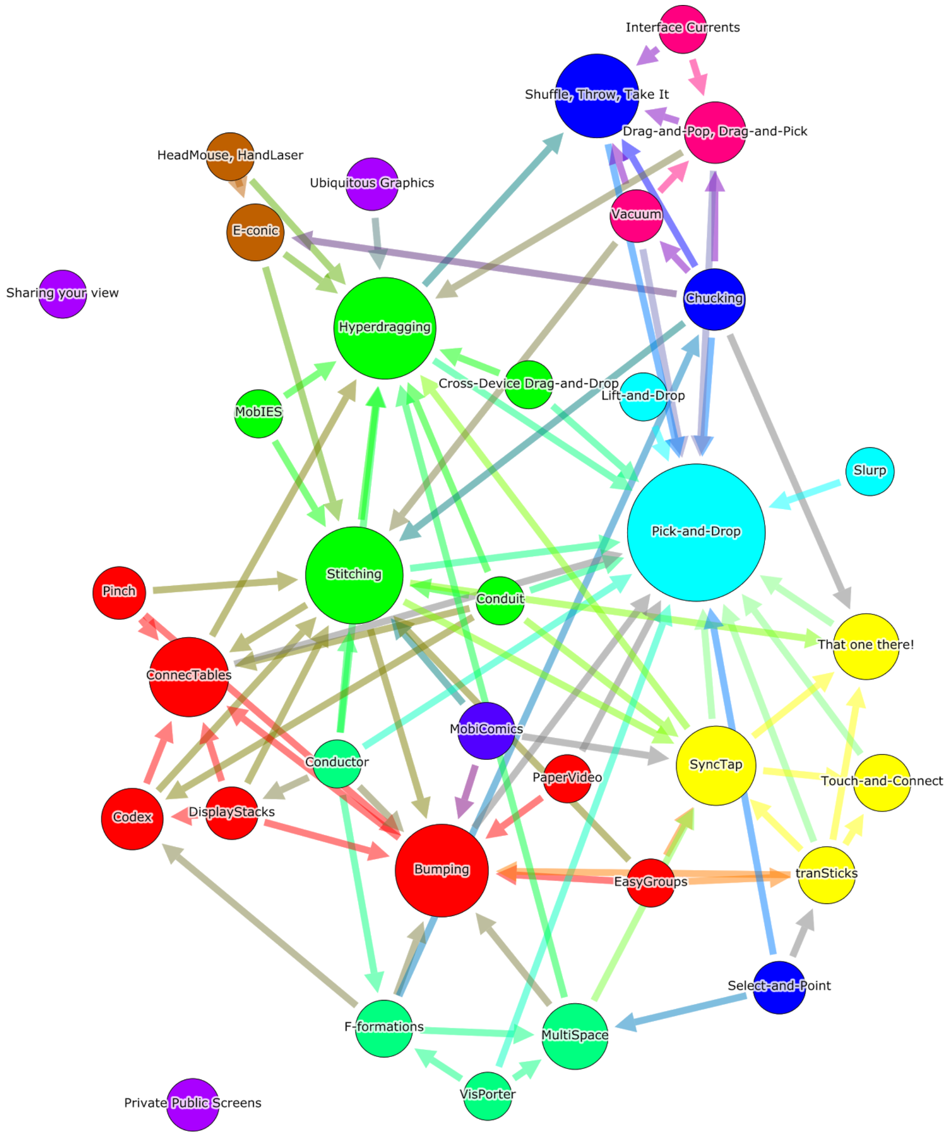


Fig. 5. Articles and citations as organized through ForceAtlas2

The algorithm was allowed to run for a few minutes, until the repulsion forces cooled down completely. The same two articles drifted away from the main cluster. Some groups are better defined than on Yifan Hu while others are more poorly defined. In this regard, they rate very similarly. Repulsion forces, though, operate in a much more acceptable fashion in ForceAtlas2: strongly connected nodes are still clustered closer together, but the more edges crossing the same area, the more repulsion applies, allowing even edge-heavy regions of the graph to be properly analyzed.

In the end, just as described by Jacomy et al. (2014), ForceAtlas2 married the clustering capacities of Yifan Hu to the force balance of Fruchterman-Reingold, but it still failed to give us meaningful understanding of the taxonomy of the articles. It became apparent that we needed to move to a deeper taxonomic analysis, instead of simply mapping all citations.

2.3.2. Refining the Taxonomic Analysis

We decided to analyse how articles cited each other, separating citations into two groups:

- **citations in passing**: the authors analyzed the existing literature on the subject, but the article described something different from what they were aiming at;
- **influential material**: the authors based one or more characteristics of their design on a previous article or their work was a continuation of research described in the previous article.

These criteria satisfy two defining tenets of a design language: it deepens our understanding of *derivation*, *aggregation*, and *association* as described by Dearden & Finlay (2006) and it encompasses the generative principle from Fincher & Windsor (2000) by analysing how one pattern creates further patterns.

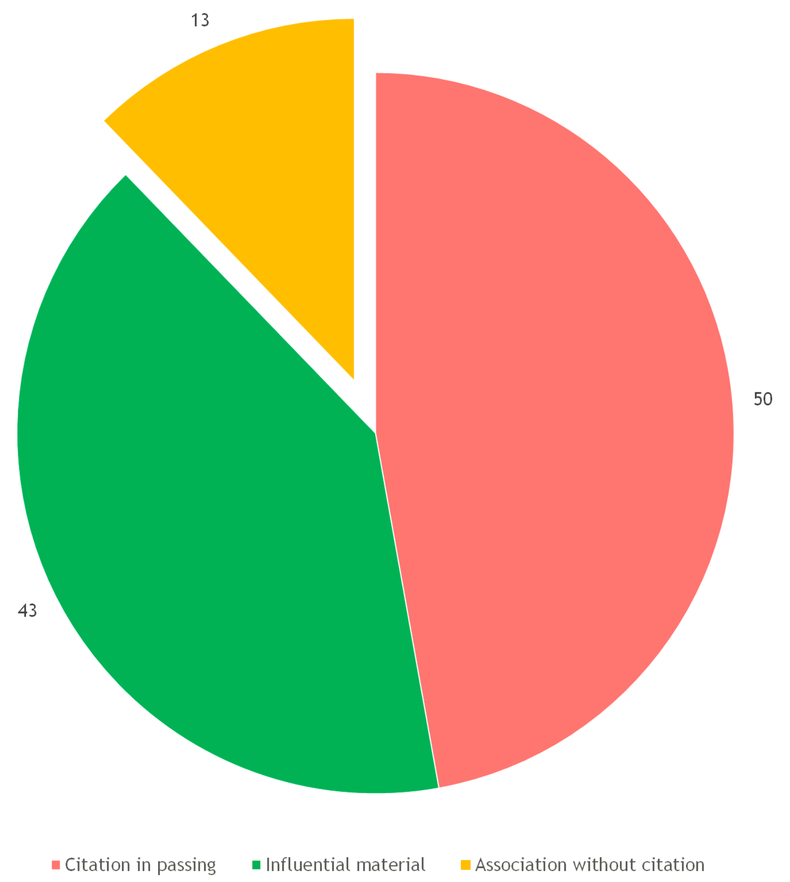


Fig. 6. Citations in the corpus by relevance

Of the 93 citations identified in our taxonomic survey, 53.76% were deemed citations in passing and 46.24% were considered true influential material. Additionally 13 instances of *association without citation* - situations where two interaction techniques were clearly related, but did not cite each other or any related article in the corpus - were identified, mostly situations where research was being done in parallel and reached similar results. Fig. 6. illustrates the proportions of each type of citation.

These citations were then weighted by relevance - 1.0 for *influential material*, 0.5 for *association without citation* and 0.3 for *citation in passing*, and fed into Gephi for analysis.

A table new relationship matrix can be seen in Fig. 7.

	1	2	3	4	5	6	7	8	9	10	13	14	15	17	20	21	23	24	25	27	30	31
2	✓																					
3	❝	✓																				
4	❝		❝																			
5	✓																					
6	❝	❝	❝																			
7	❝			✓																		
8	❝		✓		✓				✦													
9	✓							✦														
10	✓		❝	❝	❝		✓	✓														
12		❝				❝																
13	❝	❝				✓				❝												
14	❝				✓		✓	✓	❝													
15			❝				❝	❝														
16			❝																			
17			❝							❝												
18	✓											❝	✓									
19	✓																					
20	❝	✓			✓	❝				✓	❝			❝								
21				✓						✓												
22	✓																					
23							❝						✓		✓	❝						
24	✓		❝	❝				✓		✓						✓						
25				❝			✓			✓						✓						
26							✓	❝		❝		❝										
27		✦					✓	❝		❝					✦						✦	
28			❝	✦						❝												
29	❝						✓															
30				✓			✓			✓										✦		
31																				✦		
32			❝											✓								
33	❝		✓										✦									
34	❝												❝		✦		❝	✦				
35	❝		❝				✓			✓			✦				✓		❝			
36																						✦

Fig. 7. Rows represent citing papers, columns represent cited papers. [❝] are citations in passing, [✓] are citations with significant contributions to the citing paper,[✦] are associations without citation. For brevity, article names are suppressed: numbers represent articles as they appear in section 7.

A fairly straightforward grouping structure arose from running ForceAtlas2 on this weighted relationship matrix, allowing us to identify not only what articles were central concepts, but also which ones gravitated around these. The divisions were so clear-cut that very few articles were included in more than one group, all of them describing groups of patterns instead of a single one. The result of the ForceAtlas2 analysis is illustrated in Fig. 8.

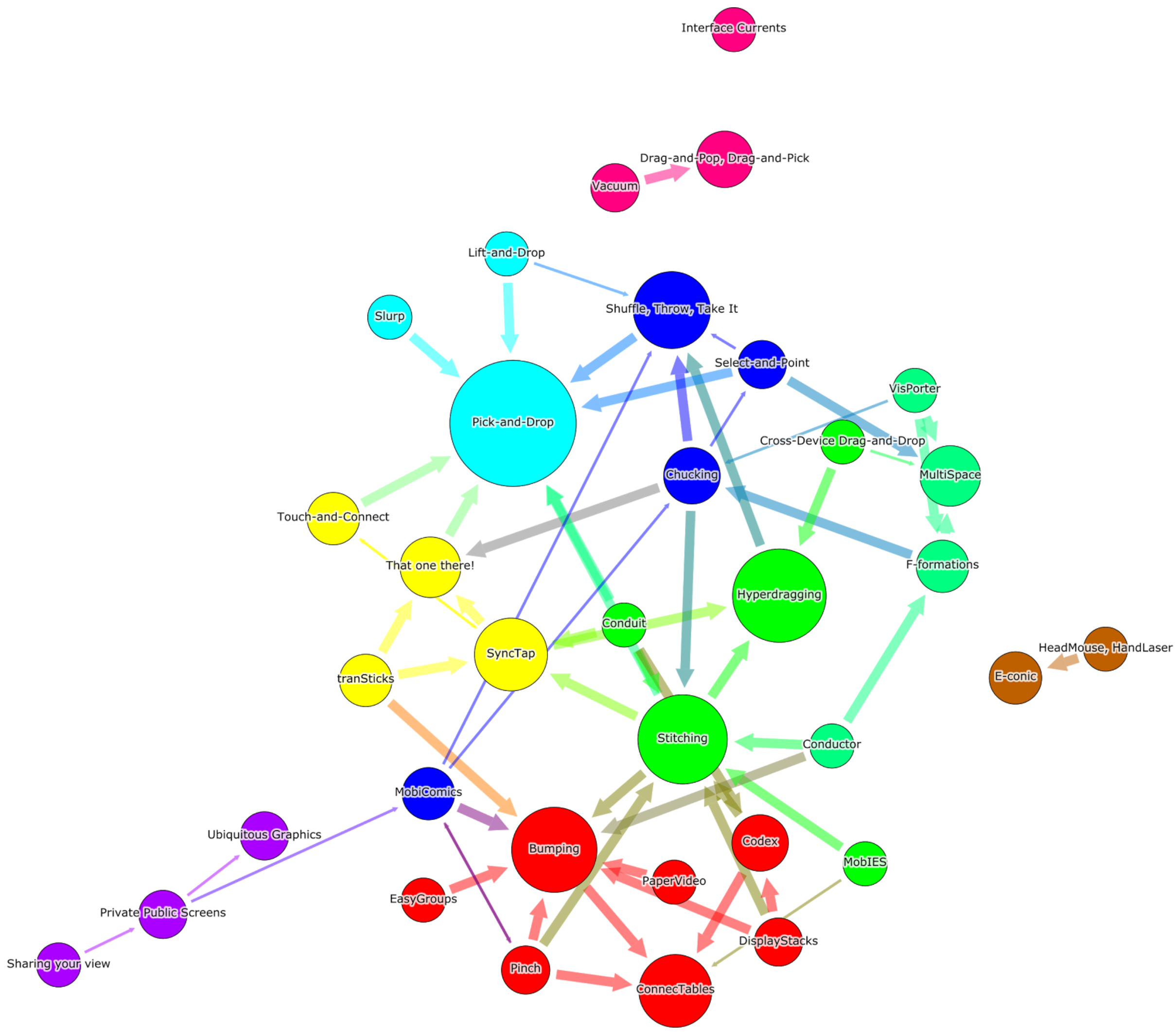


Fig. 8. ForceAtlas2 on the new relationship matrix.
Edge weight is visible in arrow scale, weights lower than 0.5 hidden for clarity.

This new weighted approach led to a change in the data presented in the pattern catalogue as three new fields were added to the input form:

- **Cites:** All articles in the corpus cited by the one where the technique is described;
- **Cited by:** All articles citing it, regardless of the weight;
- **Related to:** Citations of influential material to or from said article, plus any instance of *association without citation*.

These fields were retroactively added to the patterns already in the wiki (an example of this can be seen in **section 2.1.1**), therefore enabling Fincher & Windsor's (2000) principle of hypertextual navigability between patterns in a language.

The results of the social analysis can be found in the **Annex** in **section 5**. Moreover, all the data generated during this analysis can be found in the wiki at `http://idlab.tlu.ee/patterns/`

2.4. Grouping Patterns

Once the patterns and their relationships have been mapped, it became possible to group them according to their connections and the similarities between them. These families represent the recurring problems we are faced time and again when designing Distributed User Interfaces and, by delineating them, we not only answer our research question, but we also give some grammatical sense to how patterns relate to each other. Very few patterns belong to more than one family at the same time - these are emphasized in *italics*.

Nine major families were identified.

2.4.1. Pick-and-Drop

All patterns in this family share a common theme - instead of attempting a cross-device gesture, they use a physical proxy - a stylus, an eyedropper, or the user's own hands to transfer files between devices, a common problem when designing Distributed User Interfaces. Depends on a central server to capture the picking gesture, remembering what objects were picked by where and then arranging the transfer of said objects to the receiving device.

Seminal article: Pick-and-drop (Rekimoto, 1997)
Other members: *Shuffle, Throw or Take It* (Geißler, 1998), Slurp (Zigelbaum et al. 2008), Lift-and-Drop (Bader et al. 2010)

2.4.2. Throw

Patterns in this family make use of a common gesture - mimicking the physical throwing of an object across devices. This pattern was dormant for ten years, from the moment Geißler described it in 1998 as a method to replace long-distance dragging to the moment Lee used cameras to identify pointing in 2008. More recently accelerometers on the sending devices have been used to identify the throwing (or chucking) gesture. A different take on the problem of sending data across devices in DUIs.

Seminal article: *Shuffle, Throw or Take It* (Geißler, 1998)
Other members: Select-and-point (Lee et al., 2008), Chucking (Hassan et al., 2009), MobiComics (Lucero et al., 2012)

2.4.3. Cross-Device Dragging

Contrary to Geißler's throwing things around, Rekimoto in 1999 described a pattern for continuously dragging objects between several networked devices through the use of camera tracking for positioning devices spacially. The pattern took flight in 2004 when Hinckley used his previously described synchronous gestures to bind devices in a continuous dragging operation - most follow up patterns descend from this development. Yet another take on the problem of transferring data between devices.

Seminal article: Augmented Surfaces (Rekimoto & Saitoh, 1999)
Other members: Stitching (Hinckley et al., 2004), Conduit (Chen et al., 2012), MobiES (Schneider et al., 2012), Cross-Device Drag-and-Drop (Simeone et al., 2013)

2.4.4. Display Grouping

The largest family overall brings together patterns for joining two or more devices into a single working group, a very common problem in Distributed User Interfaces. Tandler in 2001 used special sensors aligned to the edges of mobile displays to detect connection, but further patterns use synchronous gestures, such as bumping devices together, pinching between two screens and even camera tracking of devices for grouping.

Seminal article: ConnecTables (Tandler et al., 2001)
Other members: Bumping (Hinckley, 2003), Codex (Hinckley et al. 2009), DisplayStacks (Girouard et al. 2012), EasyGroups (Lucero et al., 2012) , PaperVideo (Lissermann et al., 2012), Pinch (Ohta & Tanaka, 2012)

2.4.5. Wireless Physicality

A family that also describes patterns to joint two devices, but differentiates itself from the Display grouping patterns by the use of a physical proxy for the joining gesture: Swindells used in 2002 a special pointing stick bound to a device and capable of reading other device identifiers. Later patterns use buttons on the devices themselves to specially bound memory sticks.

Seminal article: That one there! (Swindells et al., 2002)
Other members: SyncTap (Rekimoto et al., 2003), Touch-and-Connect (Iwasaki et al., 2003), tranSticks (Ayatsuka & Rekimoto, 2005)

2.4.6. Gravity-Like

Not all relationships are born out of success: the defining characteristic between the patterns in this family is their discontent with the imprecision of Geißler's throwing gesture (all patterns were described before Lee resurrected throwing using cameras in 2008). The solutions vary slightly, but all use gravity-like forces to either attract targets closer to the dragged object or to move objects around like a stream, in a different attempt to solve the issue of moving data around a DUI.

Seminal article: Drag-and-Pop and Drag-and-Pick (Baudisch et al., 2003)
Other members: Interface Currents (Hinrichs et al., 2005), The Vacuum (Bezerianos & Balakrishnan, 2005)

2.4.7. Cross-Device Portals

Instead of demanding a cross-device dragging gesture to transfer data between devices, the patterns in this family resort to portals, visual proxies of other devices in the group, through which objects can be transported. First described by Everitt in 2006 in a table-centric interface that had fixed portals representing other displays in the group, it was further expanded, mostly through the use of proxemic patterns, to make dynamic use of portals that only present themselves when other devices are around. Also included is the concept turned on its head, where a device broadcasts an intent to share to all devices in the group for one or more recipients to accept.

Seminal article: MultiSpace (Everitt et al., 2006)
Other members: F-formations (Marquardt et al., 2012), VisPorter (Chung et al., 2014), Conductor (Hamilton & Wigdor, 2014)

2.4.8. Private Public Screens

Patterns in this family combine a shared, public screen displaying coarse information with private screens where this information can be handled in depth - by pointing the private device to a portion of the public screen, it captures the information and displays it in more detail than is being shared publicly. Sanneblad used a digital whiteboard's sonic tracking pens in 2006 for this purpose, but later implementations use other forms of near-field radio (NFC, RFID) or visual (QR-Codes, image interpretation) patterns to the same effect. This tackles a problem not addressed on other groups - how to successfully represent different levels of abstraction on the same data in separate displays.

Seminal article: Ubiquitous Graphics (Sanneblad & Holmquist, 2006)
Other members: VideoWall (Baldauf et al., 2012), Shared Views (Díez et al., 2014)

2.4.9. Perspective-Aware

These patterns compensate for user position relative to one or more devices when drawing interface elements. Nacenta used camera tracking in 2006 to locate users in space and draw objects relative to their visual position, then project these onto screens in the environment to allow for better readability and easier interaction. It was further explored in 2013 when a group of different patterns were tailored and tested for these perspective-aware displays. It, again, tackles a problem most families do not preoccupy themselves with - how to maintain readability in Multi-Display Environments.

Seminal article: E-conic (Nacenta et al., 2007)
Other members: Seamless Interaction (Chernicharo et al., 2013)

2.5. Recurring Gestures

While each of these patterns can be added to one or more families, they sometimes contain gestures that occur time and again, even in patterns of different families. The following is a list of gestures that occur in one or more patterns, sometimes under different names:

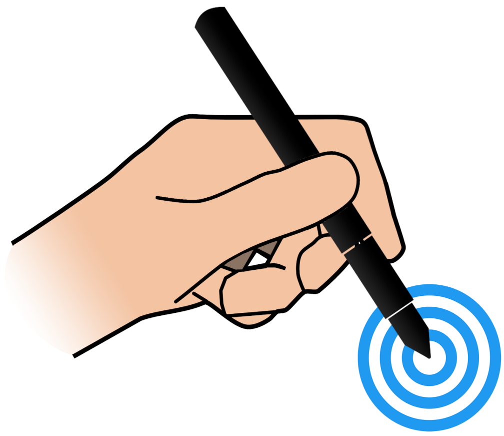


Fig. 9. **Tapping** - pressing the mouse button, your finger or the tip of a stylus against an object, then releasing. Also called **clicking**, **touching** and **selecting**.

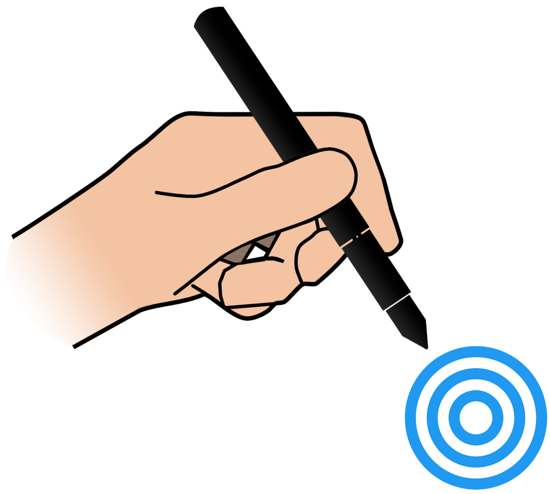


Fig 10. **Pointing** - using the tip of a wand, stylus or your finger to indicate an object without actually touching it. Depending on the technology behind the detection may need to be done from a certain distance (sometimes called **hovering**)

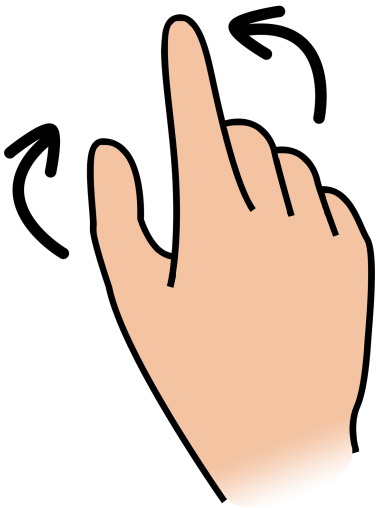


Fig 11. **Pinching** - moving two fingers towards one another. Both fingers can be on the same device or on different devices, pinching towards borders that are touching each other.

Fig 12. **Throwing** - sending an object from one place to another by mimicking the act of throwing it. Originally done by dragging towards you, dragging away from you and then dropping. This is sometimes called **flicking**. Later appropriated to mean selecting an object on a device and then mimicking throwing the device towards the target (also called **chucking**).

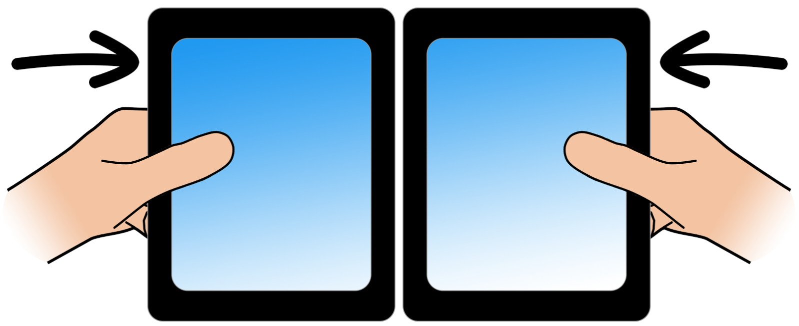


Fig. 13. **Bumping** - bringing two devices physically together. Can be done side against side (**collocation**), corner against side, corner against corner with different results.

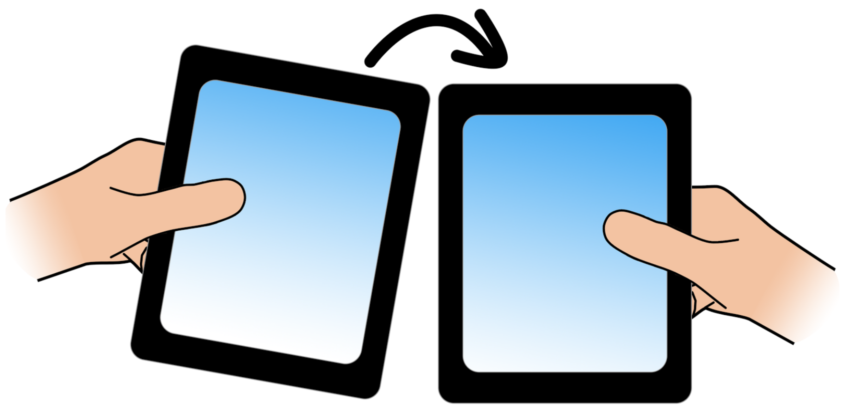


Fig 14. **Tilting** - angling your device (usually 10 degrees or more) in relation to another device. Tilting towards a device can be a gesture of sending or sharing content (called **pouring**) and tilting away from a device can be a gesture for pulling content (called **retrieving**). Tilting a device so its screen faces another device's screen can be a gesture for mirroring content (called **facing**).

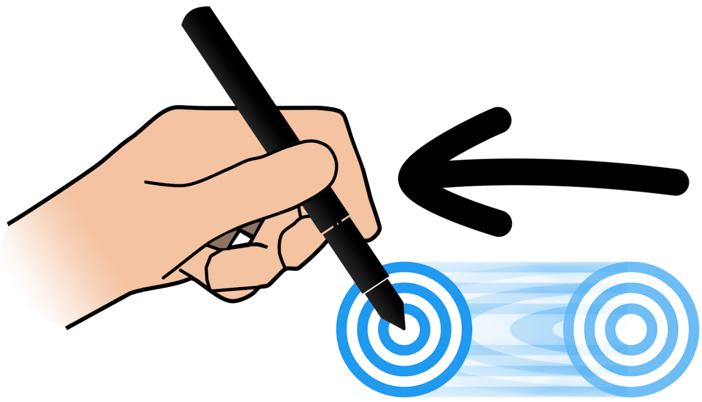


Fig 15. **Dragging** - pressing the mouse button, finger or stylus tip against an object and moving it in a certain direction, usually followed by **dropping** (releasing the button or lifting the stylus or finger). When done across screen borders can be called **stitching**. **Dropping** can be called **popping** when the targets are under influence of gravity-like effects, as in drag-and-pop - these targets will then pop back to their original places.

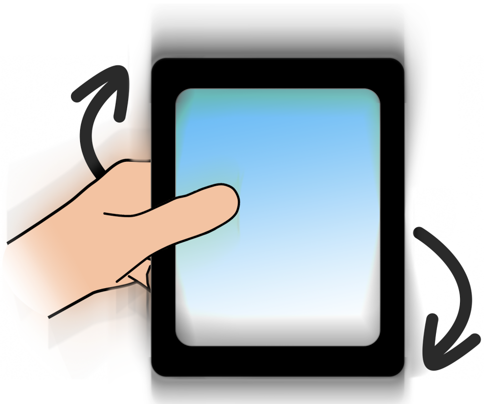


Fig. 16. **Flipping** - turning a device's screen 180 degrees from facing upwards to facing downwards or vice-versa. An extreme example of tilting, but not done in relation to other devices. Also used to cancel actions.

Fig. 17. **Stacking** - putting several objects on top of each other. When disorganized, can be called a **pile**. Like a stack of cards, can also be *linearly overlapping* or spread like a *fan*.

Fig. 18. **Shaking -** holding an object and moving it in short, rapid sideway movements. Usually done to cancel an action or delete content.

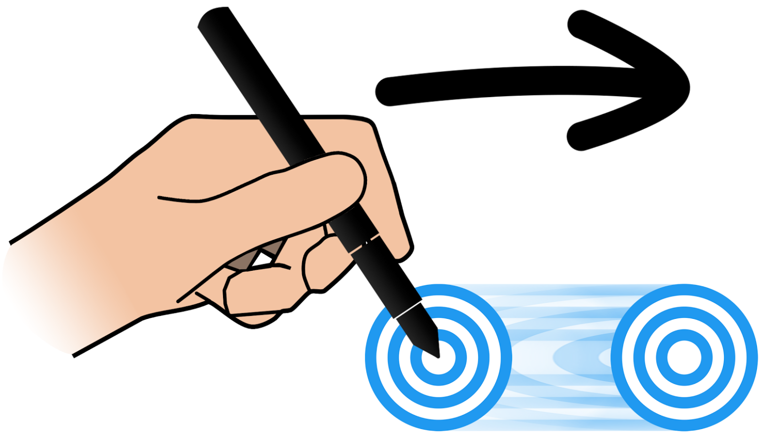


Fig 19. **Shuffling** - dragging sideways across an object's handle in order to move it sideways. A special case of throwing, dormant for over a decade.

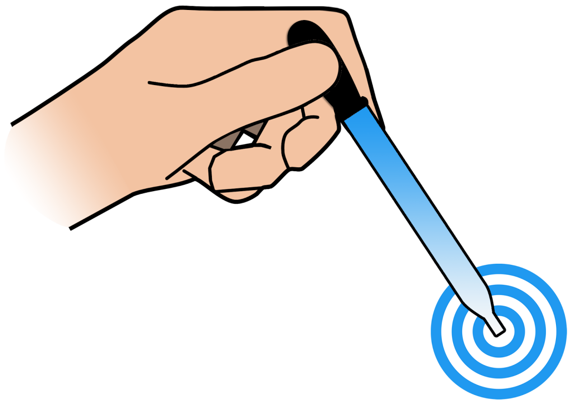
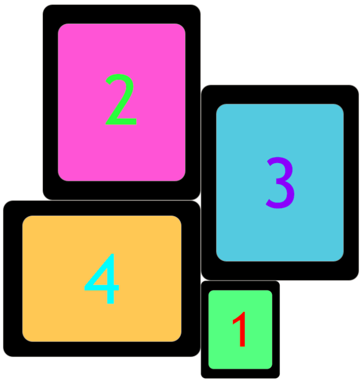


Fig 20. **Picking -** the act of removing an object from a device to later drop it somewhere else, also called **taking**, **extracting** and (air)**lifting**.

2.6. Implicit and Explicit Micro-patterns

Through the processes of grouping patterns and identifying common gestures that occur time and again in Multi-Display Environments, we were able to identify what we have termed micro-patterns, that is, the minimum indivisible actions that enable the patterns present in our catalogue. It is by applying or combining these micro-patterns into practical interaction models that the patterns in our catalogue emerge. It is these recurring patterns that represent proven solutions to the DUI design problems identified in the analyzed corpus.

While some of these micro-patterns are simply recurring, *explicit* user gestures as described in **section 2.5**, others happen *behind the scenes* without explicit user input. These *implicit micro-patterns* are no less important in understanding how patterns emerge and relate. The *implicit micro-patterns* we identified are:

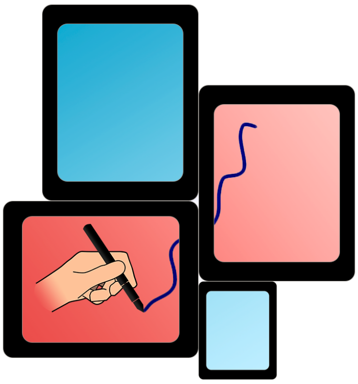


2.6.1. Screen Mapping

A pattern relates to this micro-pattern every time it needs awareness of how many screens are present in an MDE and where they are present absolutely in space or relative to each other. This is one of the most basic micro-patterns in any pattern involving more than one screen.

2.6.2. Screen Merging

A derivative of **Screen Mapping**, it's applied every time a user interface temporarily or permanently spans more than one screen. It's enabled by **Screen Mapping** and is the main enabler of the **Display Grouping** pattern group.



2.6.3. Synchronous Gestures

A rather complex concept hiding behind the simplicity of its results, it is a micro-pattern that's applied every time a gesture begins at one device and moves into another without them being previously mapped together. By identifying that two synchronous gestures in different devices are actually a single gesture spanning two devices, it is possible to infer these devices are contiguous and apply **Screen Mapping** to them.

2.6.4. Family-based Micro-patterns

The following micro-patterns emerged from analysing how other patterns grouped together: they mostly share a name with their family:

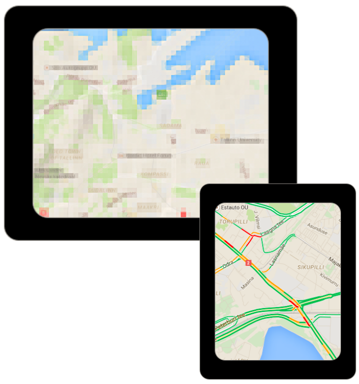
2.6.4.1. Portals

This micro-pattern is applied every time a pattern uses a proxy to represent one device on the interface of another design. The main enabler of **Cross-device**

portals, this micro-pattern can be combined with a multitude of gestures such as flicking or drag-and-dropping to allow remote interaction with or data transmission to a device without interacting directly with it.

2.6.4.2. Gravity-Like

This micro-pattern applies to patterns where not all forces being applied to objects are being exerted by the user. These “nature-like” forces: gravity, magnetism, attraction, flow, etc. allow interface components to move around without explicit, direct control of users. Main enabler of **Gravity-like** patterns.



2.6.4.3. Private Public Screens

Whenever a private device is used to see details of data presented in a coarser form on a public display is an instance of this micro-pattern. Very closely related to Augmented Reality in that what you see through your private device is an augmentation of what others see on public displays. Enables the **Private Public Screens** family.

2.6.4.4. Perspective-Aware Interfaces

These DUIs track the users' eyes in relation to one or more screens in order to print perspective-corrected interface elements. Makes for enhanced visibility for the tracked user when screens are positioned at different angles. Enables the **Perspective-Aware** family.

2.6.5. Explicit Micro-Patterns

Besides all the micro-patterns exposed above, some patterns depend explicitly on common, recurring gestures: **pointing**, **dragging**, **dropping**, **picking** and **throwing**. When two or more patterns depend on one such gesture, we've also mapped them as micro-patterns.

3. Pattern Language

By mapping out micro-patterns, their relationships with pattern groups, patterns within groups and the relationships between these we ended up with a final taxonomy that, we believe, satisfies Alexander et al.'s (1977) definition of a pattern language and Fincher & Windsor (2000) principles for a HCI pattern language:

- It allows *analyzing a pattern from several standpoints* - by motivation, design goal, setting, underlying technologies, family, relation to other patterns, etc;
- It provides for *hypertextual navigation and exploration* between patterns by linking to all upstream and downstream related patterns;

- It provides the tools to *expand and build upon it* thanks to the use of a Semantic MediaWiki to collect and organize the patterns;
- It *follows a taxonomy* which can be reproduced and expanded upon.

By dividing the map into generations, we believe we have also provided a taxonomy where all patterns are related either by *derivation*, *aggregation* or *association*, the principles delineated by Dearden & Finlay (2006) to qualify HCI pattern language taxonomies.

The taxonomy of the resulting pattern language can be seen in Fig. 21.

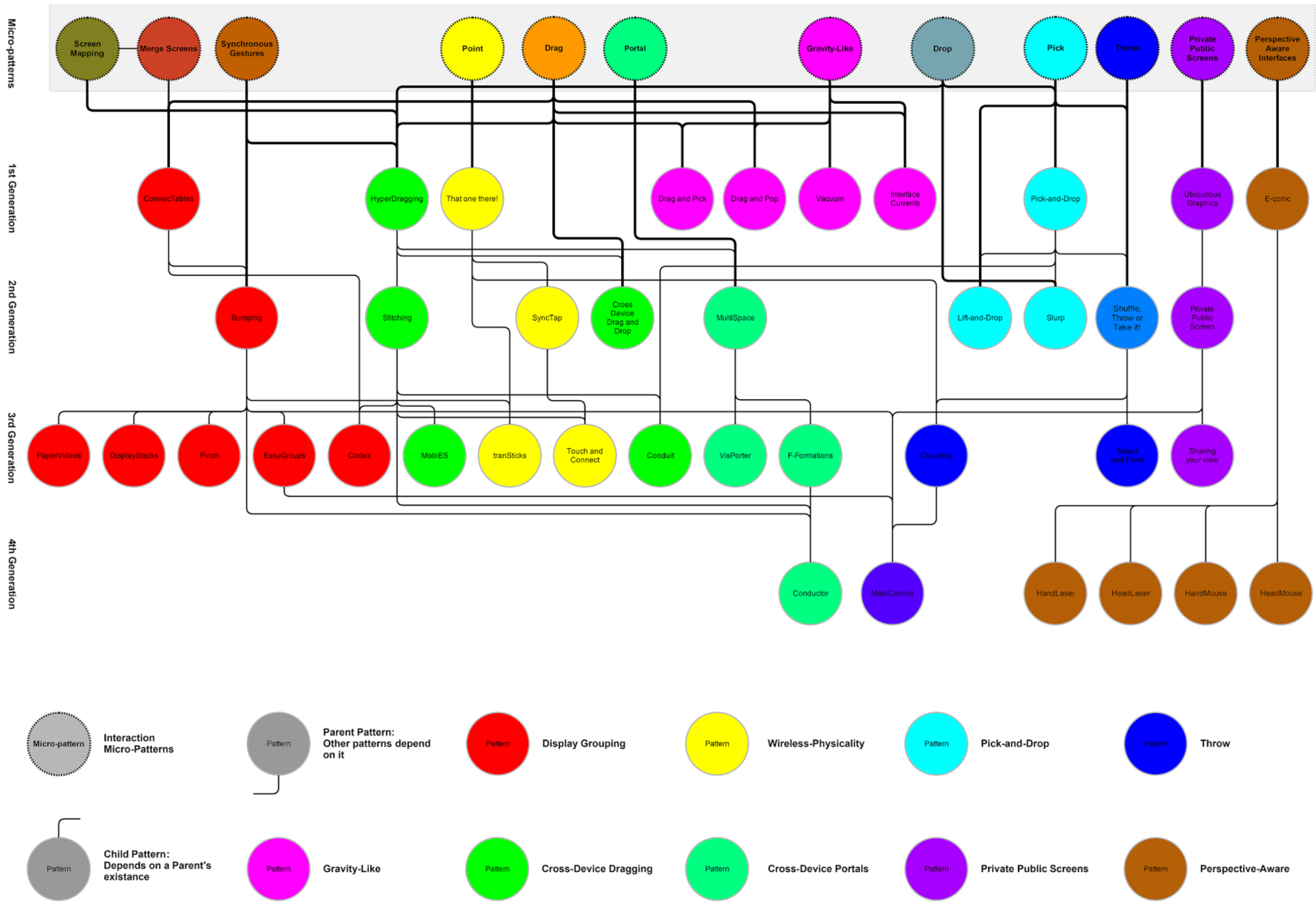


Fig. 21. Taxonomy of a Pattern Language for Distributed Interactions

4. Discussion

We set out from a research problem of figuring out how to provide interaction designers with a way of bridging the gap between design concept and tangible artifact. Our goal was, from a given corpus of distributed interaction research, build a catalogue of software design patterns as prescribed by the Group of Four (Gamma et. al, 1994). This was complemented by a secondary goal of building the scaffolding of a pattern language out of that catalogue as envisioned by Alexander (1997). The criteria used to gauge the success of this goal were the ones outlined by Dearden & Finlay in 2006 to differentiate pattern catalogues from pattern languages in the field of HCI: navigability, explorability, expandability and the existence of a taxonomy - we believe we have achieved all four.

In the process, we tried to answer two research questions:

- What are the recurring problems in distributed interaction design?
- Which patterns represent proven solutions to these identified problems?

To answer these questions, we attempted to group our patterns into families, then further reduce these into atomic *micro-patterns* which represented the minimal versions of the interaction patterns being described. These represent problems that appear time-and-again in multi-display environments: how to identify all screens present in an environment, how to connect these into collaborative interaction areas, how to move objects between screens, how to interact with objects too far away to manipulate directly, how to maintain readability in non-optimally positioned screens etc.

By understanding how patterns interconnect and how they group under these atomic *micro-patterns*, we propose a taxonomy of primary and derivative solutions to these commonplace problems of MDEs. Not all patterns are a catch-all solution to these problems: instead, they are each particular solutions to special-cases of these recurring problems.

One shortcoming of our work is that it limited itself to the corpus of 35 articles identified by Shmorgun & Lamas (2014) as describing interaction techniques. It is possible that some meaningful connections between these articles and other sources external to these 35 could eschew our taxonomic understanding of how they are organized. For example, nine of these articles cite i-LAND (Streitz et al. as cited in Tandler et al., 2001 and others), but we have not explored either i-LAND or how meaningful these nine connections are.

Another shortcoming is that, contrary to many other pattern languages, we have decided to refrain from stating a confidence rating for each pattern. While widely used - so much so that Finchley (as cited in Dearden & Finlay, 2006) made it part of his Pattern Language Markup Language - it is not one of the *essential* characteristics of HCI patterns as identified by Dearden & Finlay (2006).

Usually confidence is a subjective understanding from the part of the pattern editor on how well the solution fits a problem - an expedient we would rather avoid for lack of a solid scientific base. And while we do have a fairly decent authority measure in the form of an article's In-Degree - its number of citations within the analyzed corpus - there is a very high correlation between an article's In-Degree and its age ($\rho = 0.743$), as illustrated on the Fig. 22. As it is not essential and too error-prone, we felt it was better left out for the time being.

As a way of validating our findings, we compared our taxonomy to that of another recent article analyzing many of the same interaction techniques as the ones included in our pattern language. Rädle et al. (2015) divides user-defined gestures into two separate families: spatially aware and spatially agnostic. While it is not an instantly obvious fit within our taxonomy, some similarities emerge: most of existing techniques categorized by them as spatially aware are within or closely related to the **Screen Mapping** micro-pattern (especially the **Display Grouping** and **Cross-Device Dragging** families) while spatially agnostic techniques are closely related to the **Portal** micro-pattern. Moreover, it utilizes **Synchronous Gestures** as one of their taxonomic classifications.

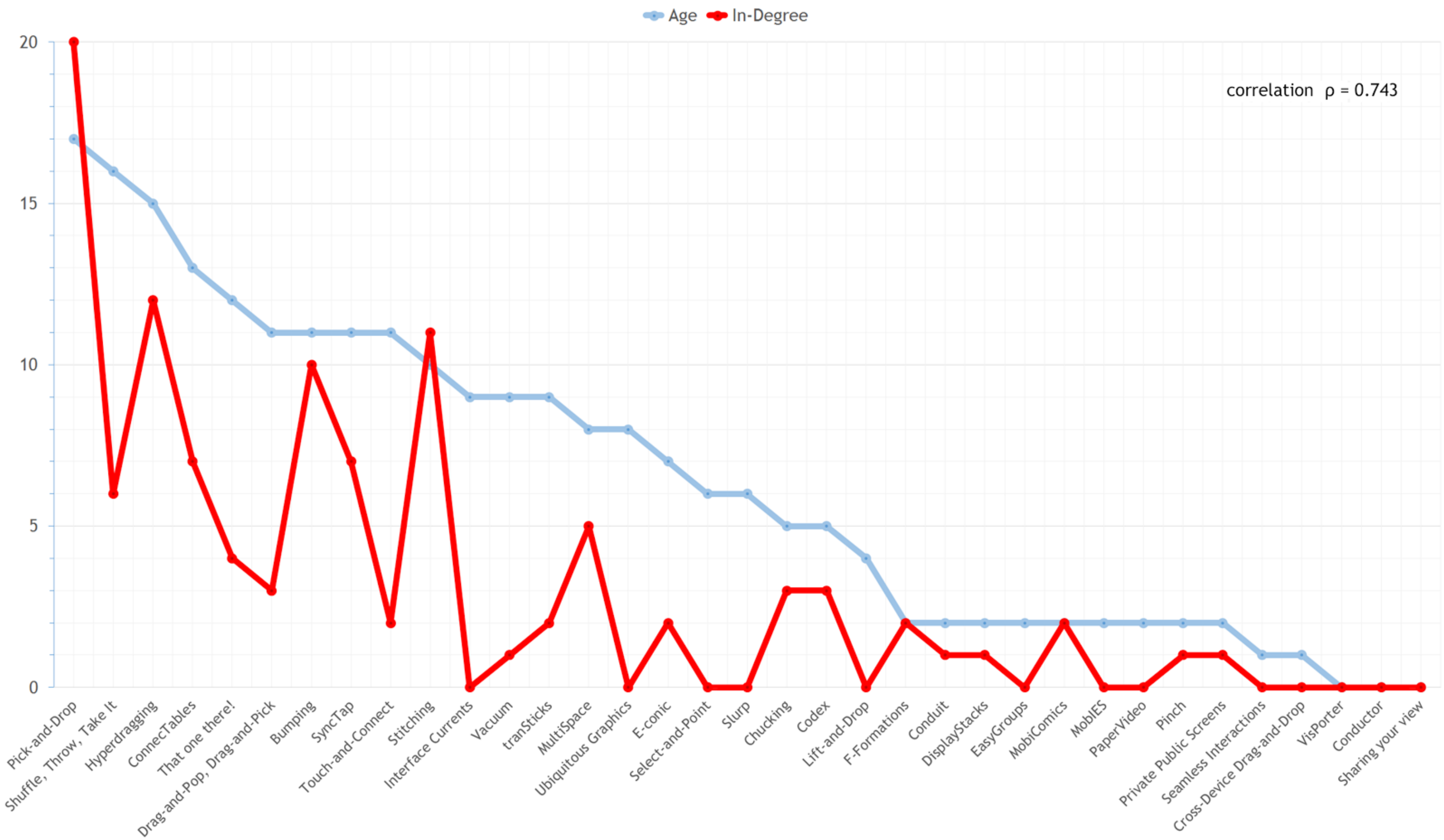


Fig. 22. Correlation between an article's In-Degree and its Age

All three techniques proposed by Rädle et al. (2015) at the final stage of their study would easily fit within our taxonomy. *Menu* is a natural fit in **Cross-device portals** and depends on very few other micro-patterns besides **Portal** itself - it would be a prototypical 1st generation **Cross-device portals** pattern. The other two techniques, *Radar View* and *Edge Bubbles*, depend on other micro-patterns, especially **Screen Mapping** as they need to position other devices spacially. They would fit in the 4th generation of **Cross-device portals**, somewhere next to *Conductor* (Hamilton & Wigdor, 2014)

5. Conclusions

We set out to synthesize a selection of Distributed User Interface interaction techniques into a pattern catalogue that could be used to facilitate future DUI designs and, in the process, developed a tentative taxonomy to classify these techniques into a DUI pattern language through social network analysis of their bibliographic citations. What we ended up with fits well with the existing definitions of Human-computer Interaction pattern languages.

We believe that by making our taxonomic exploration method explicit and easy to reproduce and by encouraging further expansion of our proposed pattern language by users of the wiki at `http://idlab.tlu.ee/patterns/` new patterns will emerge and so will evolve our understanding of this pattern language and its taxonomy but, most importantly, that future designs of Distributed User Interfaces will benefit from this resource when deciding which solutions to implement on their own design - there is an ongoing project at Tallinn University's Interaction Design Laboratory already making use of this decision tool and we hope many more will come.

While it is far from a definitive pattern language for Distributed Interactions, we believe it is a solid scaffolding the community of DUI designers can build upon and we invite them to do so.

6. Annex - *Social network analysis spreadsheet*

Id	Label	year	Out-Degree	Degree	In-Degree	Eccentricity	Closeness Centrality	Betweenness Centrality	Clustering Coefficient	Group
1	Pick-and-Drop	1997	0	20	20	0	0.000	0.000	0.100	Pick-and-drop
2	Shuffle, Throw, Take It	1998	1	7	6	1	1.000	1.100	0.238	Throw
3	Hyperdragging	1999	2	14	12	1	1.000	22.167	0.121	Cross-device Dragging
4	ConnecTables	2001	2	9	7	2	1.333	8.000	0.236	Display grouping
5	That one there!	2002	1	5	4	1	1.000	0.000	0.350	Wireless physicality
6	Drag-and-Pop, Drag-and-Pick	2003	3	6	3	1	1.000	2.333	0.300	Gravity-like
7	Bumping	2003	2	12	10	3	1.750	11.683	0.152	Display grouping
8	SyncTap	2003	4	11	7	2	1.200	31.017	0.178	Wireless physicality
9	Touch-and-Connect	2003	2	4	2	3	1.800	0.000	0.500	Wireless physicality
10	Stitching	2004	6	17	11	2	1.250	47.350	0.143	Cross-device Dragging
12	Interface Currents	2005	2	2	0	2	1.500	0.000	0.500	Gravity-like
13	Vacuum	2005	4	5	1	3	1.700	0.000	0.400	Gravity-like
14	tranSticks	2005	5	7	2	3	1.500	4.583	0.238	Wireless physicality
15	MultiSpace	2006	3	8	5	2	1.625	14.500	0.125	Cross-device Portals
16	Ubiquitous Graphics	2006	1	1	0	2	1.667	0.000	0.000	Private Public Screens
17	E-conic	2007	2	4	2	3	1.889	6.333	0.250	Perspective-Aware
18	Select-and-Point	2008	3	3	0	3	1.900	0.000	0.167	Throw
19	Slurp	2008	1	1	0	1	1.000	0.000	0.000	Pick-and-drop
20	Chucking	2009	7	10	3	3	1.500	28.100	0.167	Throw
21	Codex	2009	2	5	3	3	2.000	1.000	0.250	Display grouping
22	Lift-and-Drop	2010	1	1	0	1	1.000	0.000	0.000	Pick-and-drop
23	Tilt-to-Preview, Portals	2012	4	6	2	3	1.800	5.500	0.167	Cross-device Portals
24	Conduit	2012	6	7	1	2	1.400	3.500	0.286	Cross-device Dragging
25	DisplayStacks	2012	4	5	1	3	1.800	0.833	0.350	Display grouping
26	EasyGroups	2012	4	4	0	3	1.700	0.000	0.333	Display grouping
27	MobiComics	2012	6	8	2	2	1.571	34.000	0.143	Throw
28	MobIES	2012	3	3	0	3	1.778	0.000	0.500	Cross-device Dragging
29	PaperVideo	2012	2	2	0	4	2.200	0.000	0.500	Display grouping
30	Pinch	2012	4	5	1	3	2.000	1.000	0.417	Display grouping
31	Private Public Screens	2012	1	2	1	3	2.467	15.000	0.000	Private Public Screens
32	HeadMouse, HandLaser	2013	2	2	0	4	2.400	0.000	0.500	Perspective-Aware
33	Cross-Device Drag-and-Drop	2013	3	3	0	3	2.000	0.000	0.333	Cross-device Dragging
34	VisPorter	2014	5	5	0	3	1.765	0.000	0.200	Cross-device Portals
35	Conductor	2014	7	7	0	3	1.824	0.000	0.262	Cross-device Portals
36	Sharing your view	2014	1	1	0	4	3.313	0.000	0.000	Private Public Screens

7. References

1. Rekimoto, J. (1997). Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th annual ACM symposium on User interface software and technology* (pp. 31-39). ACM.

2. Geißler, J. (1998). Shuffle, throw or take it! working efficiently with an interactive wall. In *Conference on Human Factors in Computing Systems: CHI 98 conference summary on Human factors in computing systems* (Vol. 18, No. 23, pp. 265-266).

3. Rekimoto, J., & Saitoh, M. (1999). Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 378-385). ACM.

4. Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N., & Steinmetz, R. (2001). ConnecTables: dynamic coupling of displays for the flexible creation of shared workspaces. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (pp. 11-20). ACM.

5. Swindells, C., Inkpen, K. M., Dill, J. C., & Tory, M. (2002). That one there! Pointing to establish device identity. In *Proceedings of the 15th annual ACM symposium on User interface software and technology* (pp. 151-160). ACM.

6. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., & Zierlinger, A. (2003). Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch-and pen-operated systems. In *Proceedings of INTERACT* (Vol. 3, pp. 57-64).

7. Hinckley, K. (2003). Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (pp. 149-158). ACM.

8. Rekimoto, J., Ayatsuka, Y., & Kohno, M. (2003). SyncTap: An interaction technique for mobile networking. In *Human-Computer Interaction with Mobile Devices and Services* (pp. 104-115). Springer Berlin Heidelberg.

9. Iwasaki, Y., Kawaguchi, N., & Inagaki, Y. (2003). Touch-and-Connect: A connection request framework for ad-hoc networks and the pervasive computing environment. In *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on* (pp. 20-29). IEEE.

10. Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., & Smith, M. (2004). Stitching: pen gestures that span multiple displays. In *Proceedings of the working conference on Advanced visual interfaces* (pp. 23-31). ACM.

11. Krötzsch, M., Vrandečić, D., & Völkel, M. (2006). Semantic mediawiki. In *The Semantic Web-ISWC 2006* (pp. 935-942). Springer Berlin Heidelberg.

12. Hinrichs, U., Carpendale, S., Scott, S. D., & Pattison, E. (2005). Interface currents: Supporting fluent collaboration on tabletop displays. In *Smart Graphics* (pp. 185-197). Springer Berlin Heidelberg.

13. Bezerianos, A., & Balakrishnan, R. (2005). The vacuum: facilitating the manipulation of distant objects. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 361-370). ACM.

14. Ayatsuka, Y., & Rekimoto, J. (2005). tranSticks: physically manipulatable virtual connections. In *Proceedings of the SIGCHI conference on Human factors in computing systems*(pp. 251-260). ACM.

15. Everitt, K., Shen, C., Ryall, K., & Forlines, C. (2006). MultiSpace: Enabling electronic document micro-mobility in table-centric, multi-device environments. In *Horizontal Interactive Human-Computer Systems, 2006. TableTop 2006. First IEEE International Workshop on* (pp. 8-pp). IEEE.

16. Sanneblad, J., & Holmquist, L. E. (2006). Ubiquitous graphics: combining hand-held and wall-size displays to interact with large images. In *Proceedings of the working conference on Advanced visual interfaces* (pp. 373-377). ACM.

17. Nacenta, M. A., Sakurai, S., Yamaguchi, T., Miki, Y., Itoh, Y., Kitamura, Y., ... & Gutwin, C. (2007). E-conic: a perspective-aware interface for multi-display environments. In *Proceedings of the 20th annual ACM symposium on User interface software and technology* (pp. 279-288). ACM.

18. Lee, H., Jeong, H., Lee, J., Yeom, K. W., Shin, H. J., & Park, J. H. (2008). Select-and-point: a novel interface for multi-device connection and control based on simple hand gestures. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems* (pp. 3357-3362). ACM.

19. Zigelbaum, J., Kumpf, A., Vazquez, A., & Ishii, H. (2008). Slurp: tangibility spatiality and an eyedropper. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems* (pp. 2565-2574). ACM.

20. Hassan, N., Rahman, M. M., Irani, P., & Graham, P. (2009). Chucking: A One-Handed Document Sharing Technique. In *Human-Computer Interaction–INTERACT 2009* (pp. 264-278). Springer Berlin Heidelberg.

21. Hinckley, K., Dixon, M., Sarin, R., Guimbretiere, F., & Balakrishnan, R. (2009). Codex: a dual screen tablet computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1933-1942). ACM.

22. Bader, T., Heck, A., & Beyerer, J. (2010). Lift-and-Drop: crossing boundaries in a multi-display environment by airlift. In *Proceedings of the International Conference on Advanced Visual Interfaces* (pp. 139-146). ACM.

23. Marquardt, N., Hinckley, K., & Greenberg, S. (2012). Cross-device interaction via micro-mobility and f-formations. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (pp. 13-22). ACM.

24. Chen, N., Guimbretiere, F., & Sellen, A. (2012). Designing a multi-slate reading environment to support active reading activities. *ACM Transactions on Computer-Human Interaction (TOCHI), 19*(3), 18.

25. Girouard, A., Tarun, A., & Vertegaal, R. (2012). DisplayStacks: interaction techniques for stacks of flexible thin-film displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2431-2440). ACM.

26. Lucero, A., Jokela, T., Palin, A., Aaltonen, V., & Nikara, J. (2012). EasyGroups: binding mobile devices for collaborative interactions. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems* (pp. 2189-2194). ACM.

27. Lucero, A., Holopainen, J., & Jokela, T. (2012). MobiComics: collaborative use of mobile phones and large displays for public expression. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services* (pp. 383-392). ACM.

28. Schneider, D., Seifert, J., & Rukzio, E. (2012). MobiES: extending mobile interfaces using external screens. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia* (p. 59). ACM.

29. Lissermann, R., Olberding, S., Petry, B., Mühlhäuser, M., & Steimle, J. (2012). PaperVideo: interacting with videos on multiple paper-like displays. In *Proceedings of the 20th ACM international conference on Multimedia* (pp. 129-138). ACM.

30. Ohta, T., & Tanaka, J. (2012). Pinch: an interface that relates applications on multiple touch-screen by 'pinching' gesture. In *Advances in Computer Entertainment* (pp. 320-335). Springer Berlin Heidelberg.

31. Baldauf, M., Lasinger, K., & Fröhlich, P. (2012). Private public screens: detached multi-user interaction with large displays through mobile augmented reality. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia* (p. 27). ACM.

32. Chernicharo, J. H. D. S., Takashima, K., & Kitamura, Y. (2013). Seamless interaction using a portable projector in perspective corrected multi display environments. In *Proceedings of the 1st symposium on Spatial user interaction* (pp. 25-32). ACM.

33. Simeone, A. L., Seifert, J., Schmidt, D., Holleis, P., Rukzio, E., & Gellersen, H. (2013). Technical framework supporting a cross-device drag-and-drop technique. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia* (p. 40). ACM.

34. Chung, H., North, C., Self, J. Z., Chu, S., & Quek, F. (2014). VisPorter: facilitating information sharing for collaborative sensemaking on multiple displays. *Personal and Ubiquitous Computing, 18*(5), 1169-1186.

35. Hamilton, P., & Wigdor, D. J. (2014). Conductor: enabling and understanding cross-device interaction. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*(pp. 2773-2782). ACM.

36. Díez, D., Tena, S., Romero-Gomez, R., Díaz, P., & Aedo, I. (2014). Sharing your view: A distributed user interface approach for reviewing emergency plans. *International Journal of Human-Computer Studies*,*72*(1), 126-139.

37. Shmorgun, I., Lamas, D. (2014) The Design of Distributed User Interfaces: A Review of Literature. *Unpublished manuscript*. Tallinn University.

38. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: elements of reusable object-oriented software*. Pearson Education.

39. Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A pattern language: towns, buildings, construction (Vol. 2)*. Oxford University Press.

40. Dearden, A., & Finlay, J. (2006). Pattern languages in HCI: A critical review. *Human-computer interaction*, *21*(1), 49-102.

41. Fincher, S., & Windsor, P. (2000). Why patterns are not enough: some suggestions concerning an organising principle for patterns of UI design. In *CHI'2000 Workshop on Pattern Languages for Interaction Design: Building Momentum*.

42. Jacomy, M., Venturini, T., Heymann, S., & Bastian, M. (2014). ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PloS one*, *9*(6), e98679.

43. Melchior, J. (2011, June). Distributed user interfaces in space and time. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 311-314). ACM.

44. Elmqvist, N. (2011). Distributed user interfaces: State of the art. In *Distributed User Interfaces* (pp. 1-12). Springer London.

45. Rädle, R., Jetter, H. C., Schreiner, M., Zhihao, L., Reiterer, H., Rogers, Y., ... & Reiterer, H. (2014). Spatially-aware or spatially-agnostic? Elicitation and Evaluation of User-Defined Cross-Device Interactions. *In Proceedings of the 33rd annual ACM conference on Human factors in computing systems (CHI'15)*. ACM.